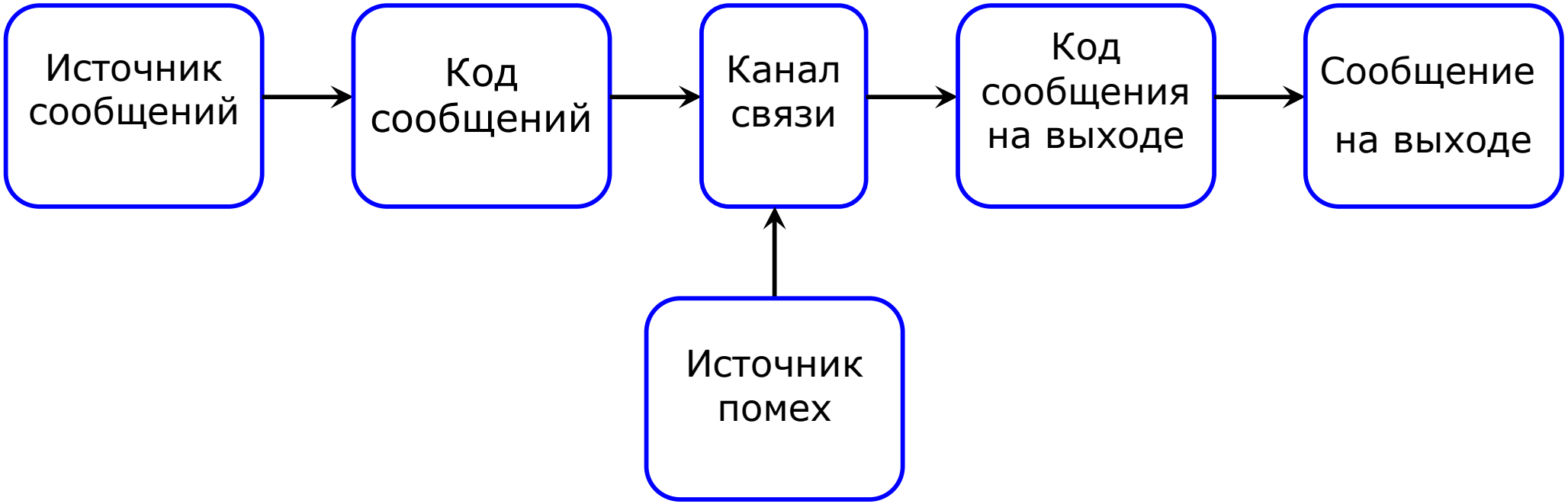
КОДИРОВАНИЕ



В этой схеме источник сообщений хочет передать по каналу связи некоторый набор *слов* — конечных последовательностей символов из заданного конечного алфавита $\mathcal{A} = \{a_1, \ldots, a_r\}$. Для передачи ему нужно (или он хочет) *закодировать* это сообщение — переписать его словами во вспомогательном алфавите $\mathcal{B} = \{b_1, \ldots, b_q\}$. После получения сообщения (возможно искаженного помехами) его нужно снова записать словами в алфавите $\mathcal{A}$ (возможно исправив возникшие ошибки).

Выбор кодов связан с различными обстоятельствами, а именно:

- с удобством передачи кодов,
- со стремлением увеличить пропускную способность канала,
- с удобством обработки кодов,
- с обеспечением помехоустойчивости,
- с удобством декодирования,
- с другими возможными требованиями к кодам.

Ниже будут рассматриваться два вида кодирования:

(а) Алфавитное кодирование. Каждой букве a_i из $\mathcal{A} = \{a_1, \ldots, a_r\}$ ставится в соответствие некоторое слово B_i из алфавита $\mathcal{B} = \{b_1, \ldots, b_q\}$. Схема кодирования, сопоставляющая эти слова, будет обозначаться буквой Σ.

(б) Равномерное кодирование. Некоторое слово B_i из алфавита $\mathcal{B}$ ставится в соответствие не букве, а какому-то слову A_i фиксированной длины в алфавите $\mathcal{A}$.

Конечно, одно из первых требований к используемому коду — требование однозначности восстановления сообщения по его коду.

Проверка однозначности декодирования

Рассмотрим алфавитные коды.

Каждое из слов $B_i,\ i=1,\ldots,r$, называется *элементарным кодом*.

Слово в алфавите $\mathcal{B}$ назовем кодовым, если его можно *расшифровать*, т.е. разбить на элементарные коды.

Одна из трудностей проверки однозначности декодирования состоит в том, что формально надо проверять бесконечное число кодовых слов.

Оказывается, этой бесконечности можно избежать.

Пусть дана схема кодирования Σ и l_i — длина слова B_i, $L = l_1 + \ldots + l_r$.

Назовем *нетривиальным разложением* слова B_i его представление в виде $B_i = \beta' B_{j_1} \ldots B_{j_w} \beta''$, где $B_{j_1} \neq B_i$, β'' является началом какого-нибудь элементарного кода, а β' является концом какого-нибудь элементарного кода. Слова β' и β'' могут быть пустыми. Набор кодовых слов $B_{j_1} \ldots B_{j_w}$ также может быть пустым.

Пример.

Σ:

$A_1 = (1\ 0\ 0\ 1)$ $\quad l_1 = 4$

$A_2 = (0)$ $\quad l_2 = 1$

$A_3 = (0\ 1\ 0)$ $\quad l_3 = 3$

Рассмотрим слово $B = 0\ 1\ 0\ 0\ 1\ 0 = A_2\, A_1\, A_2 = A_3\, A_3$

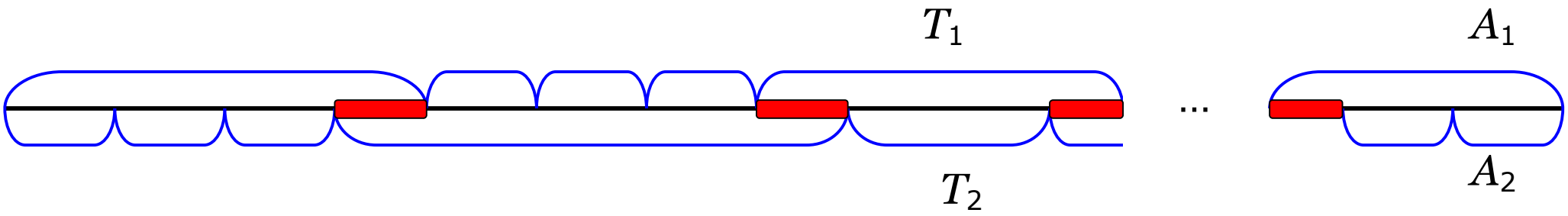
Нет однозначности декодирования!

Очевидно, что для каждого i число нетривиальных разложений слова B_i конечно. Обозначим через W максимум чисел w, взятый по всем нетривиальным разложениям всех слов B_i, $i = 1,\ldots, r$.

Теорема 1. Для любой схемы кодирования Σ найдется такое $N = N(\Sigma)$, что для проверки однозначности декодирования в Σ достаточно проверить коды слов из $\mathcal{A}$ длины не более N, и

$$N \le \lfloor (W+1)(L-r+2)/2 \rfloor.$$

Доказательство. Выберем самое короткое слово B в алфавите $\mathcal{B}$, допускающее две различные расшифровки A_1 и A_2. С ними связаны два разбиения слова B на элементарные коды T_1 и T_2 :



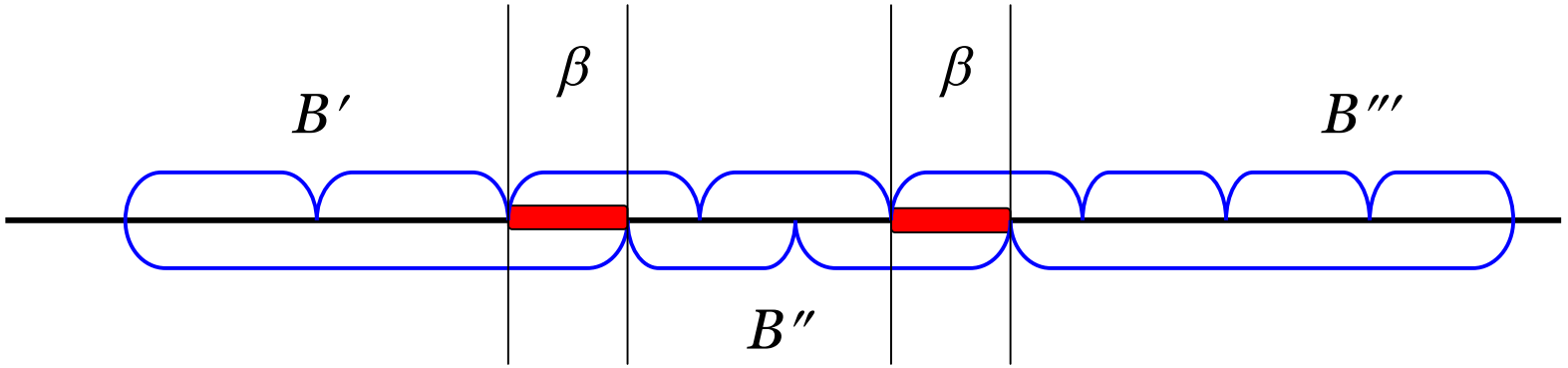
Обозначим через T разбиение, полученное после «разрезания» В там, где его «разрезало» хотя бы одно из разбиений T_1 и T_2. Части разбиения T разделим на два класса: к первому отнесем части, являющиеся элементарными кодами, ко второму — все остальные (префиксы-суффиксы).

Каждая часть β, принадлежащая второму классу, является концом одного из элементарных кодов и началом другого. Причем если β оканчивает некоторое элементарное кодовое слово в T_1, то оно начинает какое-то элементарное кодовое слово в T_2 и наоборот (см. рис.).

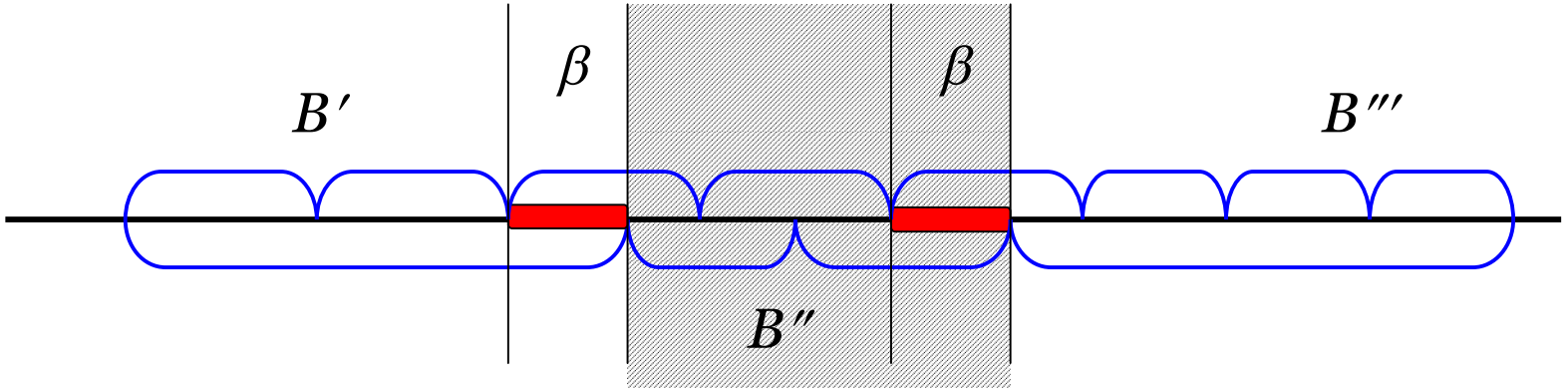
Более точно, если $B = B'\beta\ B''$, то либо $B'\beta$ и B'' являются кодовыми словами в T_1, а B' и $\beta\ B''$ являются кодовыми словами в T_2, либо наоборот.

Покажем, что все части из второго класса различны.

Допустим, что $B = B'\beta\, B''\beta\, B'''$.



Тогда слово $B'\beta\, B'''$ имеет две расшифровки в противоречие с выбором B. Чтобы убедиться в этом, заметим, что согласно вышесказанному, слова $B'\beta$, B', $\beta B'''$ и B''' являются кодовыми.

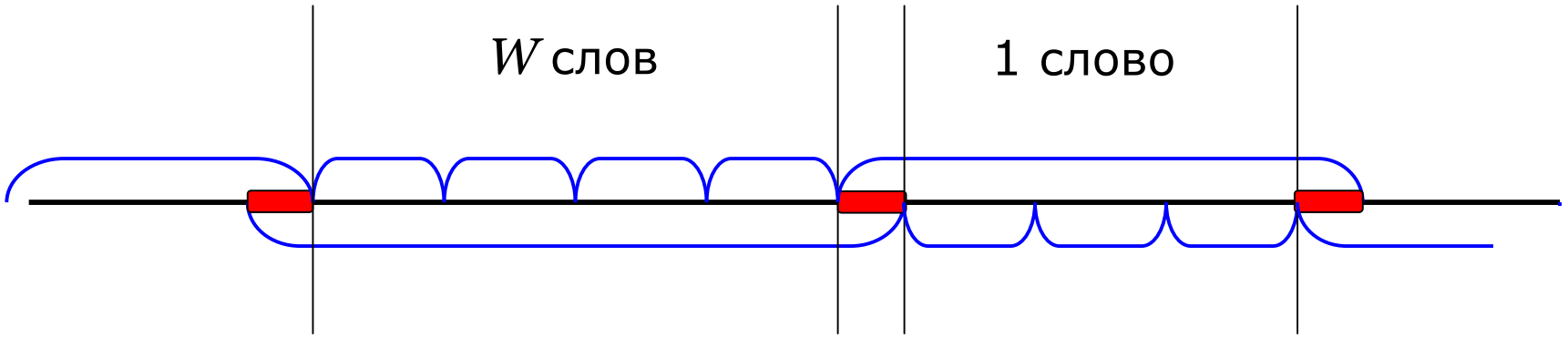


Число частиц во втором классе не превосходит числа непустых начал элементарных кодов, т.е. $(l_1 - 1) + \ldots + (l_r - 1) = L - r$.

Они дают не более $L - r + 1$ кусков.

Каждый из кусков, на которые разбивается B после выбрасывания всех частиц, является кодовым словом, входящим в одно из разбиений T_i, и частью некоторого элементарного кода, входящего в T_{3-i}.

Соседние куски являются частями элементарных кодов, входящих в различные T_i.



Два соседних куска

Имеем не более $L - r + 1$ кусков. Рассматриваем их парами.

Всего пар $\lfloor (L - r + 1)/2 \rfloor$.

В каждой паре не более $W + 1$ слов.

Следовательно, длина каждого из A_i не превосходит

$$W \cdot \lceil (L - r + 1)/2 \rceil + 1 \cdot \lfloor (L - r + 1)/2 \rfloor \le \lfloor (W + 1)(L - r + 2)/2 \rfloor . \quad \blacksquare$$

Пример.

$r = 6$, $W = 3$, $L = 20$,

$\lfloor (W + 1)(L - \mathrm{r} + 2)/2 \rfloor = \lfloor 4 \cdot 16/2 \rfloor = 32$,

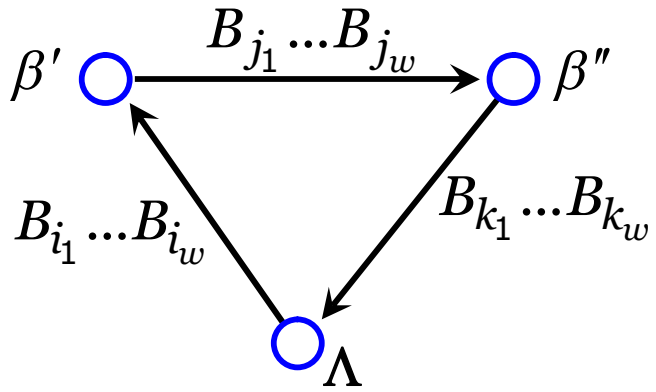
то есть требуется проверить 6^{32} слов.

Из доказательства теоремы можно извлечь существенно более эффективный алгоритм.

Пусть дана схема кодирования Σ. Для каждого элементарного кода B_i рассмотрим все его нетривиальные разложения

$$B_i = \beta' B_{j_1} \ldots B_{j_w} \beta''. \qquad (1)$$

Обозначим через $V = V(\Sigma)$ множество, содержащее пустое слово Λ и слова β, встречающиеся в разложениях вида (1) как в виде начал, так и в виде окончаний. Построим далее помеченный ориентированный граф $\Gamma = \Gamma(\Sigma)$ по следующим правилам. Множеством вершин графа Γ является $V = V(\Sigma)$. Проводим дугу из вершины $\beta' \in V$ в вершину $\beta'' \in V$, если и только если в некотором разложении вида (1) β' является началом, а β'' — концом. При этом дуга (β', β'') помечается словом $B_{j_1} \ldots B_{j_w}$.



$$B_1 = \beta' B_{j_1} \ldots B_{j_w} \beta''$$

$$B_2 = B_{i_1} \ldots B_{i_w} \beta'$$

$$B_3 = \beta'' B_{k_1} \ldots B_{k_w}$$

Теорема 2. Схема кодирования Σ не обладает свойством однозначности декодирования тогда и только тогда, когда граф $\Gamma(\Sigma)$ содержит контур, проходящий через вершину Λ.

Доказательство. Допустим, что Σ не обладает свойством однозначности декодирования. Тогда, как следует из доказательства теоремы 1, кратчайшее слово, имеющее две расшифровки в схеме Σ, имеет вид

$$B = B_{i_{1,1}} \ldots B_{i_{1,k(1)}} \beta_1 B_{i_{2,1}} \ldots B_{i_{2,k(2)}} \beta_2 \ldots \beta_{s-1} B_{i_{s,1}} \ldots B_{i_{s,k(s)}},$$

где все β_i различны и слова $B_{i_{1,1}} \ldots B_{i_{1,k(1)}}$, β_1, $\beta_1 B_{i_{2,1}} \ldots B_{i_{2,k(2)}} \beta_2, \ldots$ $\ldots, \beta_{s-1} B_{i_{s,1}} \ldots B_{i_{s,k(s)}}$ являются элементарными кодами. Это значит, что в $\Gamma(\Sigma)$ есть контур, проходящий через вершины Λ, $\beta_1, \ldots, \beta_{s-1}$.

Обратно, пусть в $\Gamma(\Sigma)$ существует контур, проходящий через вершины $\beta_0, \beta_1, \ldots, \beta_{s-1}$, где $\beta_0 = \Lambda$ и дуга (β_j, β_{j+1}), $j = 0, 1, \ldots, s-1$, $((s-1)+1=0)$, помечена словом $B_{i_{j+1,1}} \ldots B_{i_{j+1,k(j+1)}}$. Тогда слово

$$B = B_{i_{1,1}} \ldots B_{i_{1,k(1)}} \beta_1 B_{i_{2,1}} \ldots B_{i_{2,k(2)}} \beta_2 \ldots \beta_{s-1} B_{i_{s,1}} \ldots B_{i_{s,k(s)}},$$

имеет две различные расшифровки. ■

Пример. Σ: $a_1 — b_1\, b_2$

$a_2 — b_1\, b_3\, b_2$

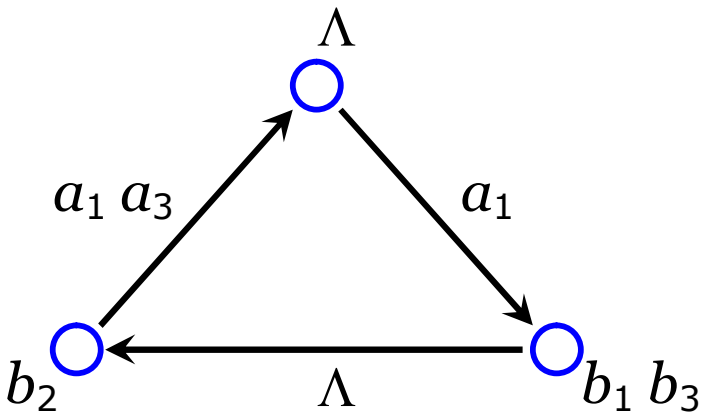
$a_3 — b_2\, b_3$

$a_4 — b_1\, b_2\, b_1\, b_3$

$a_5 — b_2\, b_1\, b_2\, b_2\, b_3$

Находим все префиксы, которые одновременно являются суффиксами и не являются кодовыми словами:

$\{\Lambda,\ b_2,\ b_1\, b_3\}$, то есть три вершины в графе



$a_1\, a_2\, a_1\, a_3 =$

$= b_1\, b_2\, b_1\, b_3\, b_2\, b_1\, b_2\, b_2\, b_3 =$

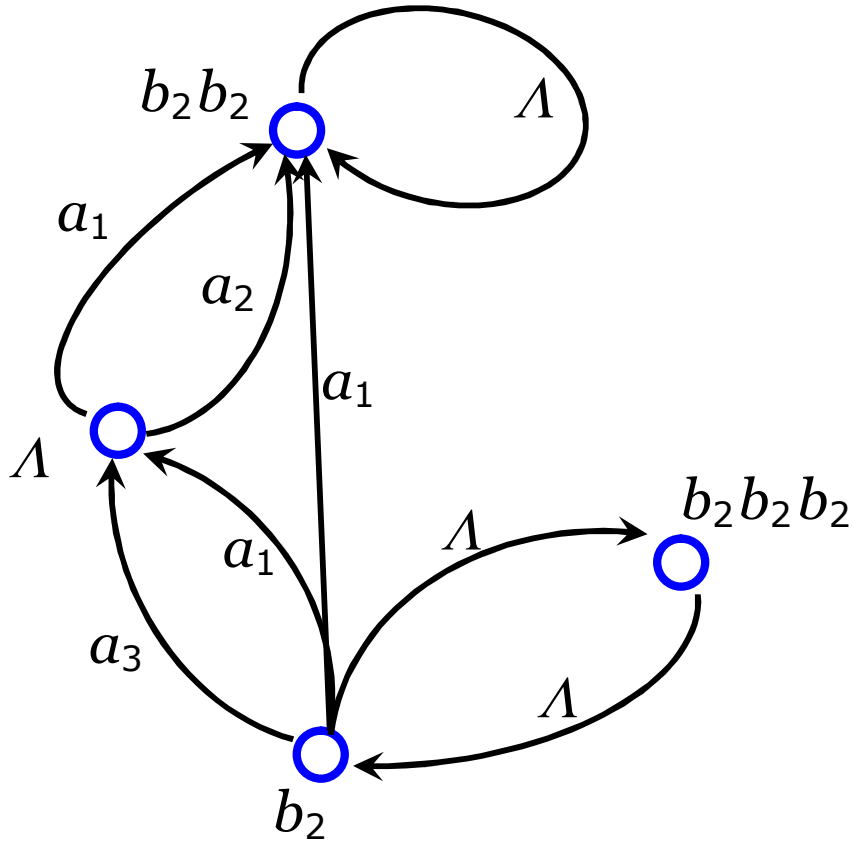
$= a_4\, a_5$

Пример.

$$\Sigma: \quad a_1 — b_1$$
$$a_2 — b_2\, b_1$$
$$a_3 — b_1 b_2\, b_2$$
$$a_4 — b_2\, b_1\, b_2\, b_2$$
$$a_5 — b_2\, b_2\, b_2\, b_2$$

Находим все β: $\{\Lambda, b_2,\ b_2\, b_2,\ b_2\, b_2\, b_2\}$

Тогда получаем граф:



Нет цикла через вершину Λ.

Код однозначно декодируется.

Префиксные коды

Важным классом однозначно декодируемых кодов являются *префиксные коды* — такие алфавитные коды, где ни один элементарный код не является *префиксом* (т.е. началом) другого элементарного кода.

Упражнение. Доказать, что любой префиксный код является однозначно декодируемым.

Обозначим через q значность алфавита, например, $q = 2$, и $l_i = l(B_i)$, $i = 1, \ldots, r$.

Теорема 3. (Неравенство Макмиллана) Если схема кодирования Σ обладает свойством однозначности декодирования, то

$$\sum_{i=1}^{r} q^{-l_i} \le 1. \qquad (2)$$

Доказательство. Выберем произвольное n. Рассмотрим коды всех r^n слов длины n в алфавите $\mathcal{A}$, полученные с помощью Σ. Все они могут быть порождены выражением

$$(a_1 + \ldots + a_r)^n,$$

если рассматривать произведение $a_{i_1} a_{i_2} \ldots a_{i_n}$ как запись слова. Имеем

$$(a_1 + \ldots + a_r)^n = \sum_{(i_1 i_2 \ldots i_n)} a_{i_1} a_{i_2} \ldots a_{i_n} .$$

Соответствующие этим словам коды получаются заменой символов $a_1, \ldots, a_r$ на элементарные коды $B_1, \ldots, B_r$. Получаем

$$(B_1 + \ldots + B_r)^n = \sum_{(i_1 i_2 \ldots i_n)} B_{i_1} B_{i_2} \ldots B_{i_n} .$$

Этому тождеству соответствует

$$\left(\frac{1}{q^{l_1}} + \ldots + \frac{1}{q^{l_r}} \right)^n = \sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}} . \qquad (3)$$

Положим $t = l_{i_1} + \ldots + l_{i_n}$ и $\nu(n, t)$ — число кодовых слов $B_{i_1} B_{i_2} \ldots B_{i_n}$ длины t. Пусть $l = \max\limits_{1 \le i \le r} l_i$. Из взаимной однозначности алфавитного кодирования вытекает $\nu(n, t) \le q^t$ и длина каждого из наших кодовых слов не превосходит nl.

Следовательно,

$$\sum_{(i_1 \ldots i_n)} \frac{1}{q^{l_{i_1} + \ldots + l_{i_n}}} = \sum_{t=1}^{nl} \frac{\nu(n,t)}{q^t} \le nl.$$

Используя (3), получаем

$$\left(\frac{1}{q^{l_1}} + \ldots + \frac{1}{q^{l_r}} \right) \le \sqrt[n]{nl}$$

Это неравенство справедливо для любого n, а его правая часть стремится к 1 при $n \to \infty$. Поскольку его левая часть не зависит от n, необходимо, чтобы $q^{-l_1} + \ldots + q^{-l_r} \le 1$ ■

Следующий факт характеризует префиксные коды с положительной стороны.

Теорема 4. Если схема кодирования Σ обладает свойством однозначности декодирования, то существует такая префиксная схема кодирования Σ', что для каждого i, i=1,... , s длина l_i' элементарного кода B_i' в Σ' равна длине l_i элементарного кода B_i в Σ .

Доказательство. Можно считать, что элементарные коды B_i занумерованы в порядке неубывания их длин. Пусть длинами элементарных кодов в Σ являются числа λ_1,..., λ_s, $\lambda_1 < \lambda_2 < \ldots < \lambda_s$ и число элементарных кодов длины λ_i , i =1,..., s равно ν_i. Тогда неравенство Макмиллана можно переписать в виде

$$\sum_{t=1}^{s} \frac{\nu_t}{q^{\lambda_t}} \le 1. \qquad (4)$$

В частности, $\nu_1 / q^{\lambda_1} \le 1$, откуда $\nu_1 \le q^{\lambda_1}$. Выберем среди q^{λ_1} слов длины λ_1 в алфавите $\mathcal{B}$ произвольные ν_1 слов в качестве элементарных кодов $B'_1,\ldots,B'_{\nu_1}$. Перейдем к словам длины λ_2. Из (4) получаем

$$\frac{\nu_1}{q^{\lambda_1}} + \frac{\nu_2}{q^{\lambda_2}} \le 1,$$

$$\nu_2 \le q^{\lambda_2} - \nu_1 q^{\lambda_2 - \lambda_1}. \qquad (5)$$

Рассмотрим множество слов длины λ_2 в алфавите $\mathcal{B}$, не начинающихся с $B'_1,\ldots,B'_{\nu_1}$. В силу (5) из этого множества можно выбрать ν_2 каких-нибудь слов в качестве элементарных кодов $B'_{\nu_1+1},\ldots,B'_{\nu_1+\nu_2}$.

Далее из (4) получаем

$$\nu_3 \le q^{\lambda_3} - \nu_1 q^{\lambda_3-\lambda_1} - \nu_2 q^{\lambda_3-\lambda_2}$$

и строим ν_3 слов длины λ_3, не начинающихся с $B'_1,\ldots,B'_{\nu_1+\nu_2}$ и т.д. Через конечное число шагов построим нужное количество слов нужной длины. По построению новый код будет префиксным. ■

Лекция 12-13

Кодирование

Коды с минимальной избыточностью

Префиксные коды

Важным классом однозначно декодируемых кодов являются *префиксные коды* — такие алфавитные коды, где ни один элементарный код не является *префиксом* (т.е. началом) другого элементарного кода.

Упражнение. Доказать, что любой префиксный код является однозначно декодируемым.

Обозначим через q значность кодирующего алфавита, например, $q = 2$, и $l_i = l(B_i), i = 1, ..., r$.

Теорема 3. (Неравенство Макмиллана) Если схема кодирования Σ обладает свойством однозначности декодирования, то

$$\sum_{i=1}^{r} q^{-l_i} \leq 1.$$

Следующий факт характеризует префиксные коды с положительной стороны.

Теорема 4. Если схема кодирования Σ обладает свойством однозначности декодирования, то существует такая префиксная схема кодирования Σ', что для каждого i, i=1,... , s длина l_i' элементарного кода B_i' в Σ' равна длине l_i элементарного кода B_i в Σ.

При выборе схемы кодирования естественно учитывать экономичность, т.е. средние затраты времени на передачу и прием сообщений.

Предположим, что задан алфавит $\mathcal{A} = \{a_1, \ldots, a_r\}$, $r \geq 2$, и набор вероятностей $(p_1, \ldots, p_r)$, $p_1 + \ldots + p_r = 1$ появлений букв $a_1, \ldots, a_r$. Тогда *избыточностью кодирования схемой* Σ называется величина

$$l_{cp} = l_{cp}(\Sigma) = l_1 p_1 + \ldots + l_r p_r,$$

т.е. математическое ожидание длины элементарного кода, l_i — длина кодового слова для a_i.

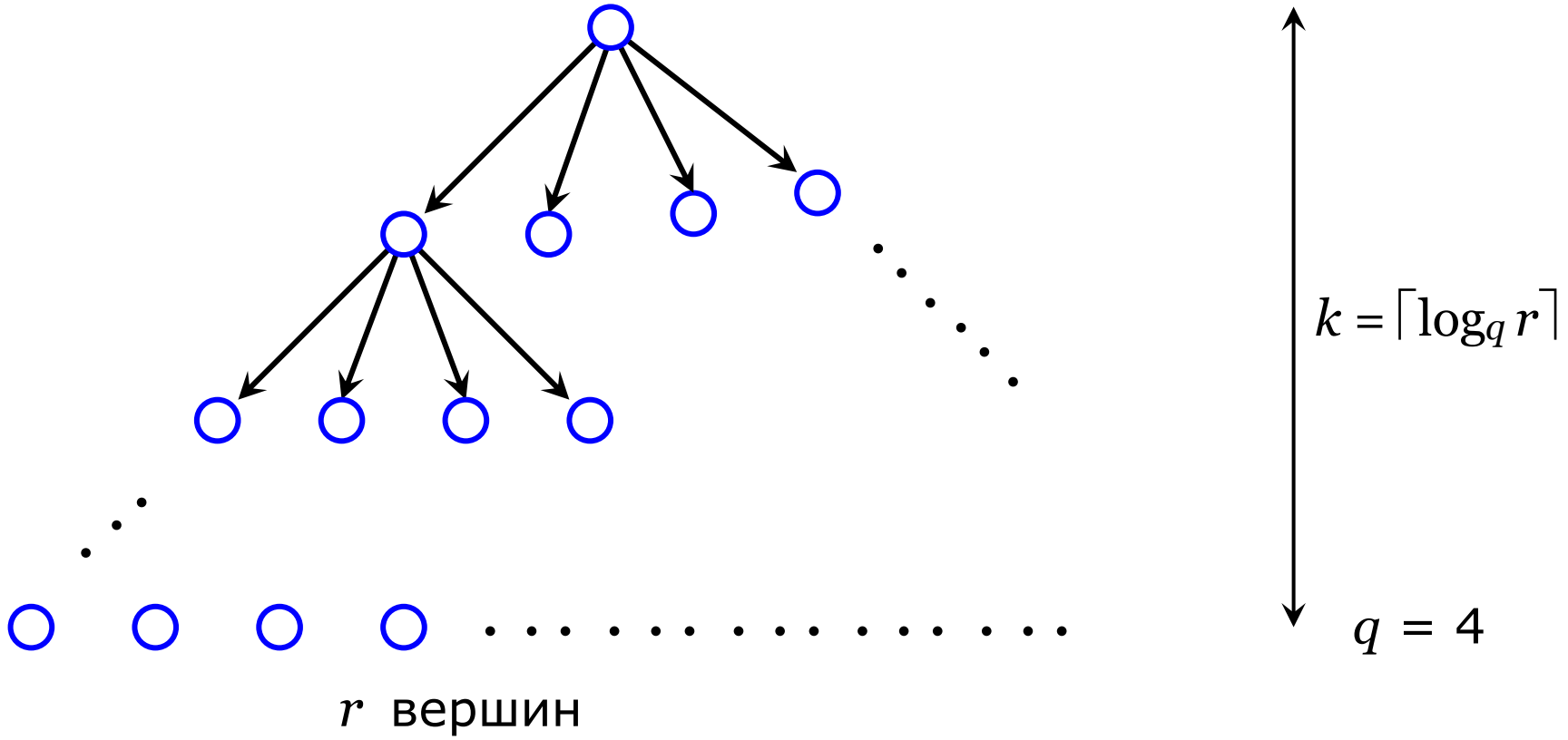
Чем меньше l_{cp}, тем экономнее в среднем схема Σ.

Пусть

$$l_* = l_*(a_1, \ldots, a_r, p_1, \ldots, p_r) = \inf l_{cp},$$

где инфимум взят по всем однозначно декодируемым схемам.

Пусть $k = \lceil \log_q r \rceil$. Тогда все a_i можно закодировать разными словами длины k в алфавите $\mathcal{B} = \{b_1, \ldots, b_q\}$. Очевидно, такое кодирование будет префиксным (а, следовательно, и взаимно однозначным). Отсюда $l_* \le k$. Таким образом, значение l_* достигается на некоторой схеме, так как для каждого i достаточно смотреть слова длины не более k/p_i, $p_i > 0$.



Коды, определяемые схемами Σ с $l_{cp} = l_*$, называются *кодами с минимальной избыточностью* или *кодами Хаффмана*. Согласно теореме 4 существуют префиксные коды с минимальной избыточностью.

Каждому префиксному коду поставим в соответствие *кодовое дерево* — ориентированное корневое дерево $T = T(\Sigma)$ по следующим правилам. Множество вершин $V(T)$ дерева T состоит из элементарных кодов и всех их префиксов, включая пустое слово. Дуга в T ведет из C в D, если C является префиксом D и короче D ровно на одну букву.

Пример 1.

a_1 — b_1b_3

a_2 — b_3

a_3 — b_1b_1

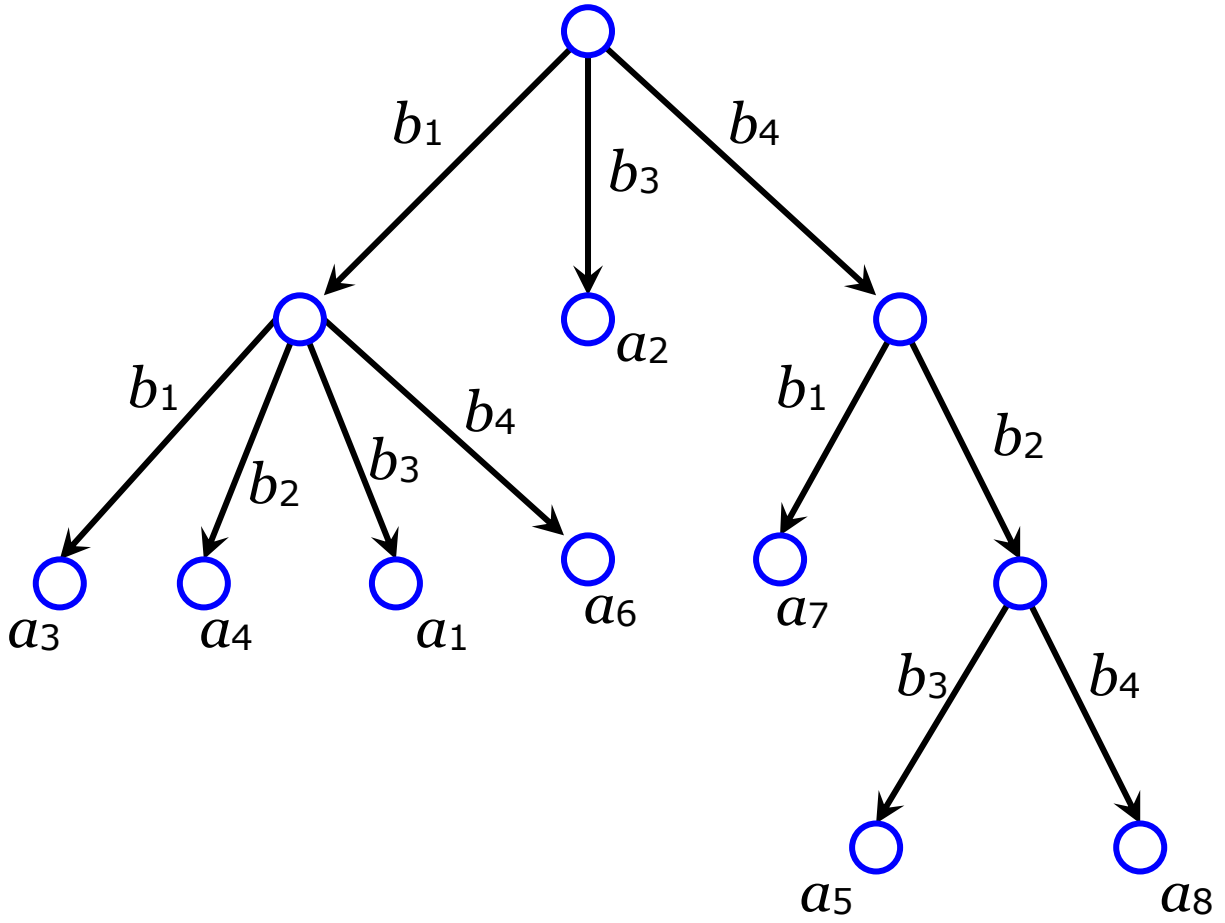
a_4 — b_1b_2

a_5 — $b_4b_2b_3$

a_6 — b_1b_4

a_7 — b_4b_1

a_8 — $b_4b_2b_4$



Элементарные коды соответствуют висячим вершинам в T, q=4.

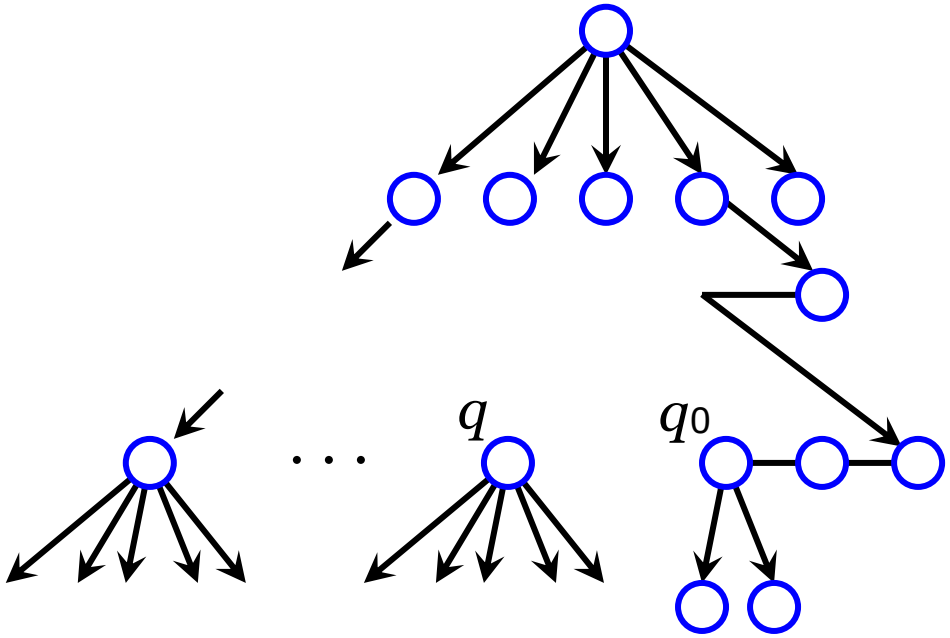
Соответствует ли T оптимальному коду при p_i = 1/8?

Итак, пусть T — кодовое дерево префиксного кода с минимальной избыточностью (со схемой Σ). Можно считать, что $p_1 \geq \ldots \geq p_r$. Тогда можно преобразовать Σ таким образом, чтобы

(а) $i < j \Rightarrow l_i \leq l_j$;

(б) порядки ветвления всех его вершин, за исключением быть может одной q_0, лежащей в предпоследнем ярусе, равны или 0, или q;

(в) порядок ветвления q_0 исключительной вершины (если она есть) не равен 1.



Если порядки ветвления всех вершин T равны или 0, или q, то положим $q_0 = q$. Ввиду (б), по индукции легко видеть, что для некоторого целого t имеем $r = t(q - 1) + q_0$. Следовательно, если h — остаток от деления r на $q - 1$, то

$$q_0 = \begin{cases} h, & \text{если } h \geq 2, \\ q, & \text{если } h = 1, \\ q-1, & \text{если } h = 0. \end{cases} \qquad (*)$$

Нетрудно видеть, что можно выбрать такой префиксный код с минимальной избыточностью, кодовое дерево которого кроме (а)–(в) обладает свойством

(г) для некоторой вершины v, лежащей в предпоследнем ярусе, порядок ветвления вершины v равен q_0, а потомками v являются $a_r, a_{r-1}, \ldots, a_{r-q_0+1}$.

Пример 2.

a_1 — b_1b_3 — 0,22

a_2 — b_2 — 0,20

a_3 — b_1b_1 — 0,14

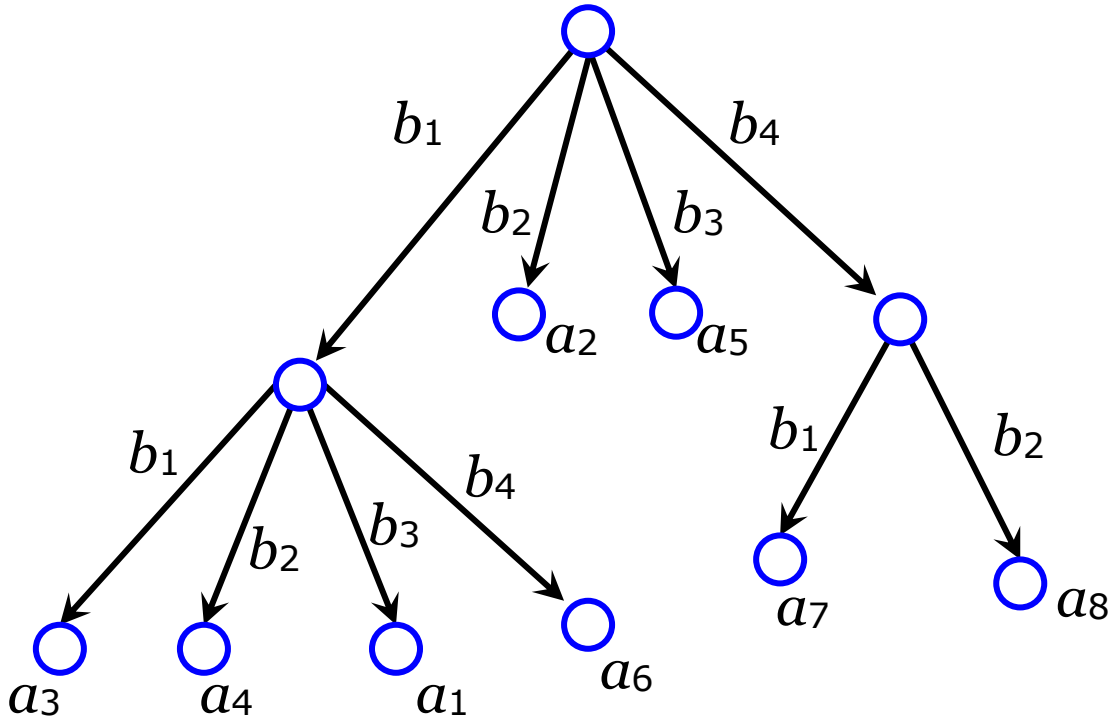
a_4 — b_1b_2 — 0,11

a_5 — b_3 — 0,10

a_6 — b_1b_4 — 0,09

a_7 — b_4b_1 — 0,08

a_8 — b_4b_2 — 0,06



$l_{cp} = 2 \cdot 0{,}22 + 0{,}20 + 2 \cdot 0{,}14 + 2 \cdot 0{,}11 + 0{,}1 + 2 \cdot 0{,}09 + 2 \cdot 0{,}08 + 2 \cdot 0{,}06 = 1{,}7$

Верно ли, что это минимум по всем однозначно декодируемым кодам?

Теорема 5. Пусть схема кодирования Σ задает код с минимальной избыточностью для алфавита $\mathcal{A} = \{a_1, \ldots, a_r\}$ и набора вероятностей $(p_1, \ldots, p_r)$, а ее кодовое дерево T удовлетворяет свойствам (а)–(г).

Обозначим $p'_{r-q_0+1} = p_r + p_{r-1} + \ldots + p_{r-q_0+1}$, а через T' — кодовое дерево, полученное из T удалением вершин $a_r, a_{r-1}, \ldots, a_{r-q_0+1}$ и сопоставлением образовавшейся висячей вершине v буквы a'_{r-q_0+1}. Тогда T' является кодовым деревом кода с минимальной избыточностью для алфавита $\mathcal{A}' = \{a_1, a_2, \ldots, a_{r-q_0}, a'_{r-q_0+1}\}$ и набора вероятностей $\{p_1, p_2, \ldots, p_{r-q_0}, p'_{r-q_0+1}\}$.

Доказательство. Обозначим схему, которой соответствует T', через Σ', номер уровня вершины v через m. Тогда

$$l_{cp}(\Sigma') = l_{cp}(\Sigma) - (m+1)(p_r + p_{r+1} + \ldots + p_{r-q_0+1}) + mp'_{r-q_0+1} = l_{cp}(\Sigma) - p'_{r-q_0+1}.$$

Если бы для алфавита $\mathcal{A}' = \{a_1, \ldots, a_{r-q_0}, a'_{r-q_0+1}\}$ и набора вероятностей $(p_1, \ldots, p_{r-q_0+1})$ нашлась схема Θ' префиксного кодирования с меньшей избыточностью чем $l_{cp}(\Sigma) - p'_{r-q_0+1}$, то подвесив в кодовом дереве для Θ' к вершине v вершины $\{a_r, a_{r-1}, \ldots, a_{r-q_0+1}\}$, получили бы схему Θ префиксного кодирования для алфавита $\mathcal{A} = \{a_1, \ldots, a_r\}$ с $l_{cp}(\Theta) = l_{cp}(\Theta) + p'_{r-q_0+1} < l_{cp}(\Sigma)$. Противоречие с выбором Σ завершает доказательство теоремы. ■

Данная теорема в сочетании с предыдущими леммами дает следующий алгоритм построения кодов с минимальной избыточностью.

Прямой ход

1. Если $r = 1$, то переходим к обратному ходу.
2. Упорядочим вероятности так, чтобы $p_1 \geq \ldots \geq p_r$.
3. Выберем q_0 по правилу (*), удалим из списка вероятностей p_r, p_{r-1},..., p_{r-q_0+1} и добавим $p'_{r-q_0+1} = p_r + p_{r-1} + \ldots + p_{r-q_0+1}$. Положим $r = r - q_0 + 1$, уберём штрих с p'_r и перейдем к шагу 1.

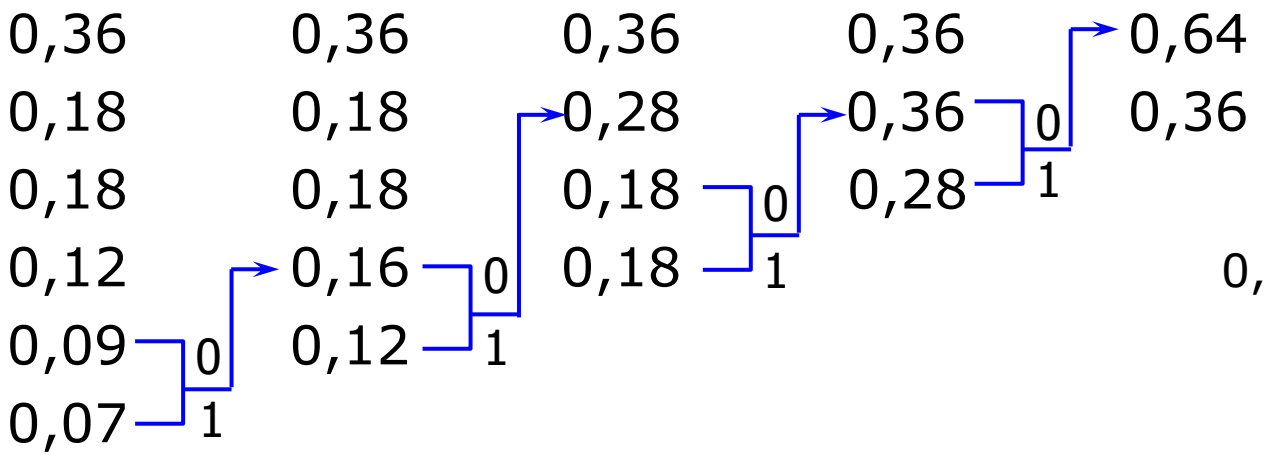
Обратный ход

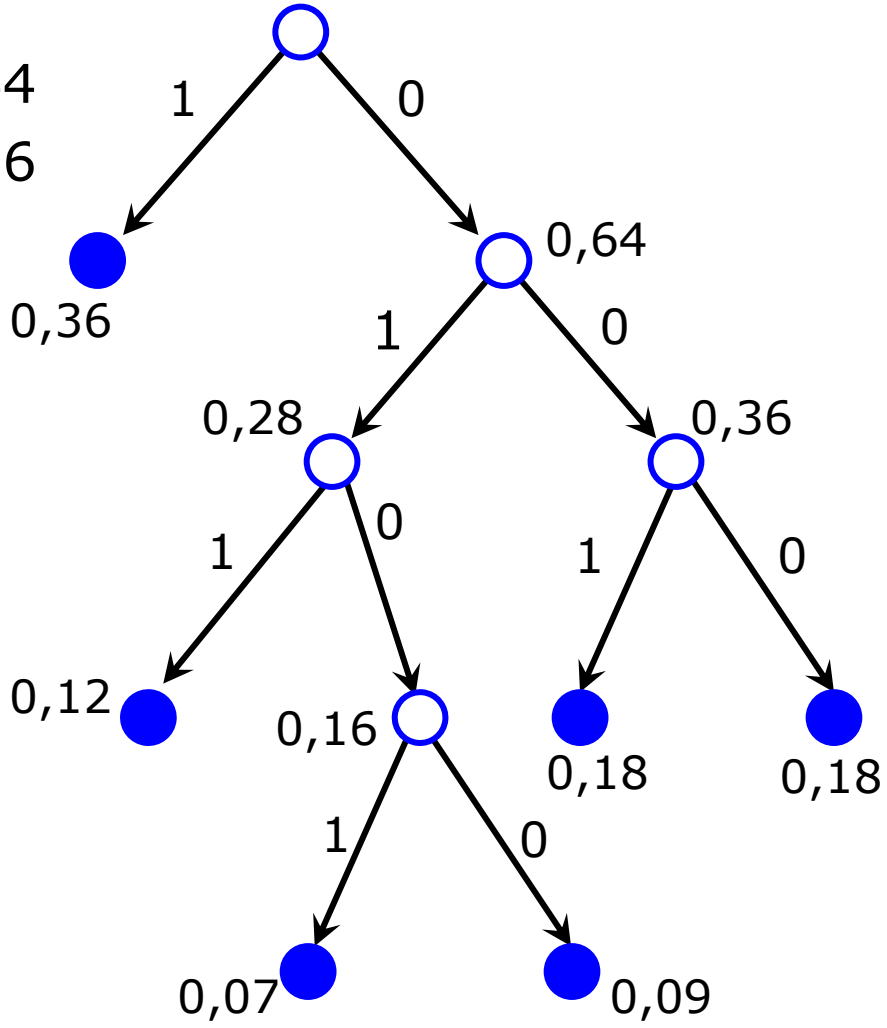
Кодовым деревом для одной буквы является одна вершина. В порядке, обратном к тому, в котором склеивались вероятности, расклеиваем вершины кодового дерева.

Пример 3.

p={0,36; 0,18; 0,18; 0,12; 0,09; 0,07}, q = 2, r = 6.

Построим код Хаффмана

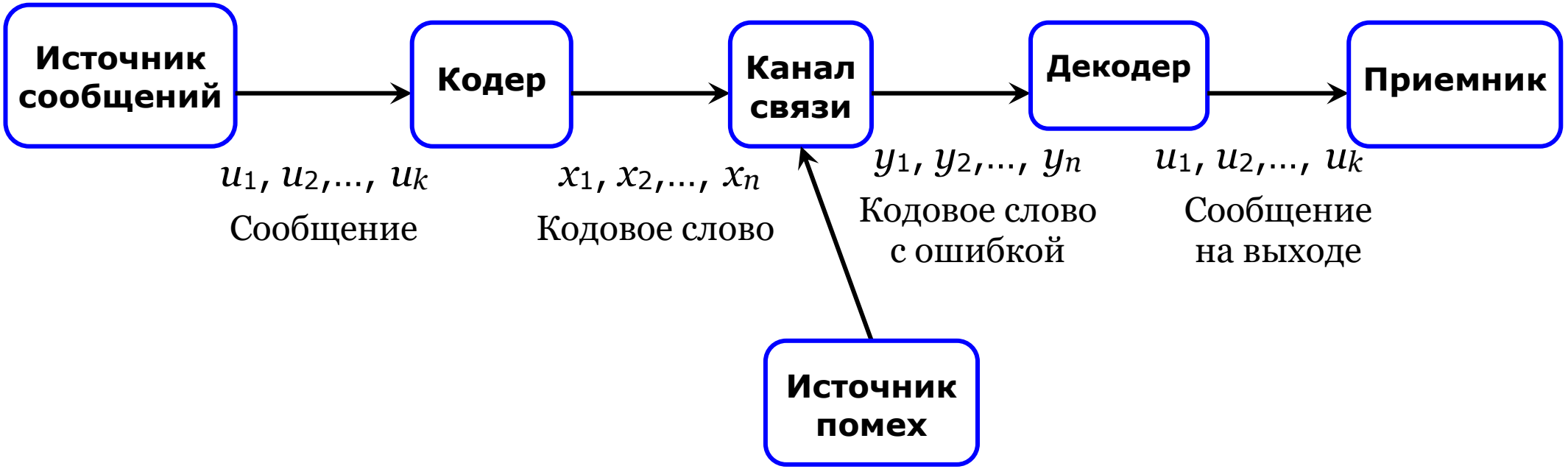




0,36 — 1
0,18 — 000
0,18 — 001
0,12 — 011
0,09 — 0100
0,07 — 0101

l_{cp} = 2,44

Кодирование сообщений



Самокорректирующиеся коды

Рассмотрим одну из простейших ситуаций, когда сообщение может искажаться в канале связи. Предположим, что в канале связи действует источник помех, который в слове длины n искажает не более p символов. Возникает вопрос: для какого m можно все m-буквенные слова в алфавите $\mathscr{A}$ закодировать n-буквенными словами так, чтобы по коду на выходе закодированные слова однозначно восстанавливались? И как это сделать?

Если $n > 2p$, то $m \geq \lfloor n/(2p+1) \rfloor$. Действительно, каждую букву можно писать $2p + 1$ раз подряд. Но такое кодирование не является самым экономным. Более того, само сообщения составляет малую часть передаваемого слова.

Код Хэмминга

В произвольном алфавите трудно восстановить исходное сообщение, даже если известен номер буквы в слове, в которой произошло искажение. Поэтому в самокорректирующих кодах обычно рассматриваются бинарные алфавиты.

Пусть x, y — бинарные слова длины n. Обозначим

$$\rho(x,y) = \sum_{i=1}^{n} (x_i \oplus y_i) = w(x-y)$$ — *расстояние Хэмминга*

Пусть B_n — Множество кодовых слов длины n. Корректировка кода состоит в выборе кодового слова ближайшего к полученному сообщению. Для того чтобы код при любых ошибках восстанавливался правильно достаточно, чтобы расстояние между любыми кодовыми словами было больше p.

Обозначим E_n^p шар радиуса p в пространстве $\{0;1\}^n$. Тогда шары с центрами в кодовых словах не пересекаются. Найдем максимальную длину m сообщений, которые можно передать n буквенными словами.

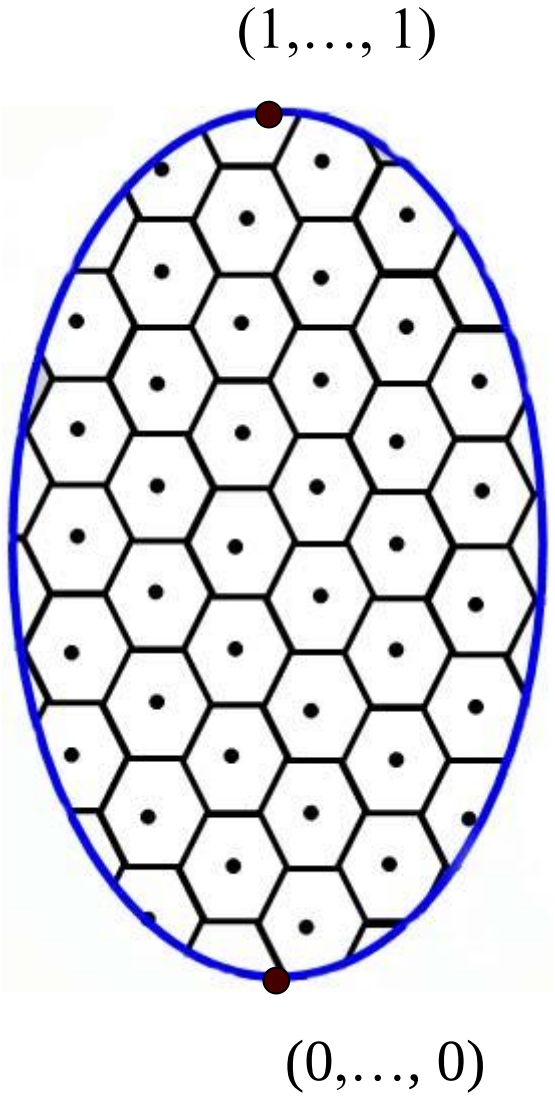
$$2^m \cdot \left|E_n^p\right| \le 2^n.$$

Откуда

$$m \le n - \log_2 \left|E_n^p\right| = n - \log_2 \left(\sum_{i=0}^{p} C_n^i \right) \le n - p \log_2 n .$$

В частности для случая одной ошибки получаем

$$m \le n - \log_2 (n+1) .$$



Пример. Пусть при передаче 15 бинарных символов происходит не более 1 ошибки. Можно ли закодировать все 13 буквенные сообщения.

Решение. $m \le n - \log_2(n+1) = 15 - \log_2 16 = 11.$

Код Хэмминга устраняющий 1 ошибку.

1. Кодирование. Пусть (a_1, a_2,…, a_m) исходное сообщение, где m максимальное значение для которого выполняется условие. Обозначим $k=n-m$.

Построим кодовое слово (b_1, b_2, …, b_n). Назовём символы с номерами $\{2^0; 2^1;\ldots; 2^{k-1}\}$ контрольными, а остальные m информационными. Поставим исходное сообщение в информационные символы.

Обозначим V_i множество индексов в двоичной записи которых на i-той позиции стоит 1. Положим $b_{2^{i-1}} = \sum_{V_i \setminus \{2^{i-1}\}} b_j \bmod 2$.

Пример. Пусть l=7. Передаваемое сообщение (1, 0, 1).

Кодировка. m=7-3=4, k=3. Контрольные символы {1; 2; 4}. Дополним исходное сообщение 0 и запишем в информационные символы (*;*;1;*;0;1;0). Найдем значения контрольных символов.

V_1={1;3;5;7}, b_1=1+0+0=1;

V_2={2;3;6;7}, b_2=1+1+0=0;

V_3={4;5;6;7}, b_4=0+1+0=1.

Кодовое слово (1;0;1;1;0;1;0).

Замечание.

$$\sum_{V_i} b_j = 0 \bmod 2 .$$

Замечание.

Контрольные символы входят в единственную сумму.

Упражнение. Доказать, что для любых кодовых слов расстояние между ними не меньше 3.

2. Обнаружение и исправление ошибок.

Пусть при передаче кодового слова (b_1, b_2,…, b_n) произошла ошибка в символе s. Пусть двоичная запись числа $s=x_k\ldots x_1$.

Найдем $\sum_{V_i} b_j' \bmod 2$. Возможны 2 случая.

$x_i=0$. Тогда $\sum_{V_i} b_j' = \sum_{V_i} b_j = 0 \bmod 2$.

$x_i=1$. Тогда $\sum_{V_i} b_j' = \sum_{V_i} b_j - b_s + b_s' = b_s' - b_s = 1 \bmod 2$.

Таким образом, контрольные суммы нарушаются только для тех номеров, которые входят в двоичную запись числа s.

Пример. Получено сообщение (1;1;1;1;0;1;0). Восстановить исходное сообщение.

Решение. Посчитаем контрольные суммы.

V_1={1;3;5;7}, x_1=1+1+0+0=0;

V_2={2;3;6;7}, x_2=1+1+0+1=1;

V_3={4;5;6;7}, x_3=1+0+1+0=0.

Искажен символ номер $s=010_2=2$.

Отправленное сообщение (1;0;1;1;0;1;0).

Замечание. Если все контрольные суммы равны 0, то полученное сообщение пришло без искажений.