Коды с минимальной избыточностью

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.

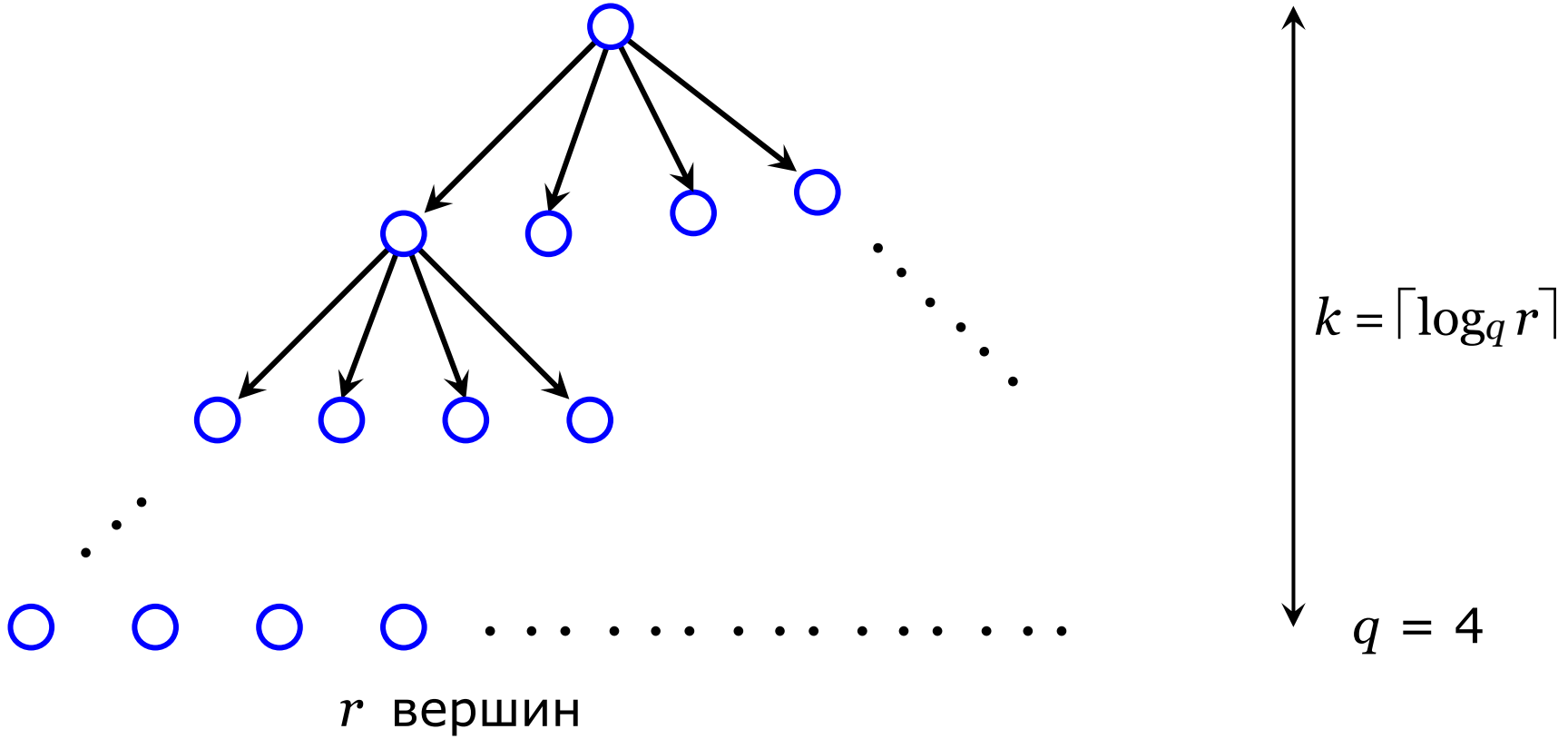
Чем меньше l_{cp}, тем экономнее в среднем схема Σ.

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 $\mathcal{B}$. 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 (см. лекцию 10) 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.

$a_1 \ — \ b_1 b_3$

$a_2 \ — \ b_3$

$a_3 \ — \ b_1 b_1$

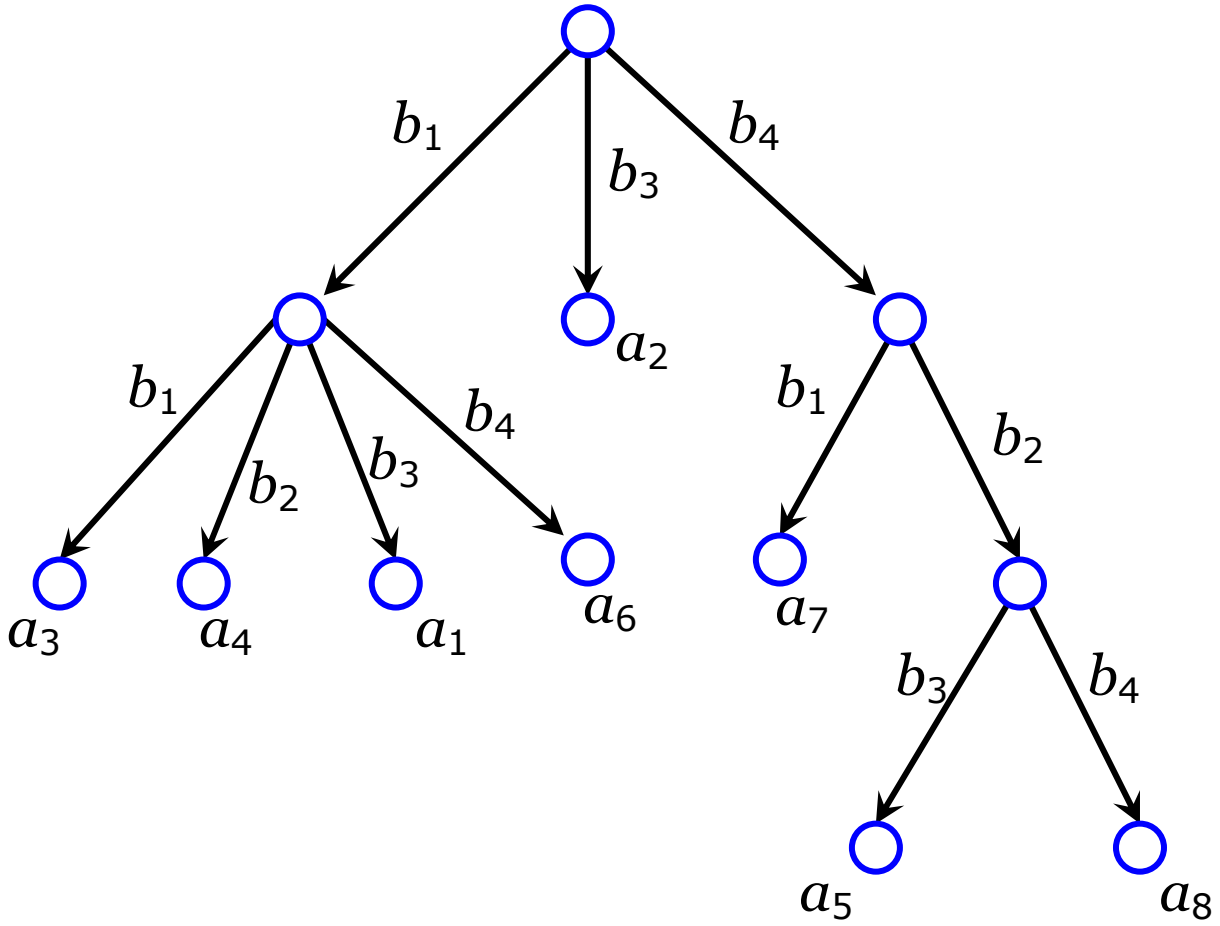
$a_4 \ — \ b_1 b_2$

$a_5 \ — \ b_4 b_2 b_3$

$a_6 \ — \ b_1 b_4$

$a_7 \ — \ b_4 b_1$

$a_8 \ — \ b_4 b_2 b_4$



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

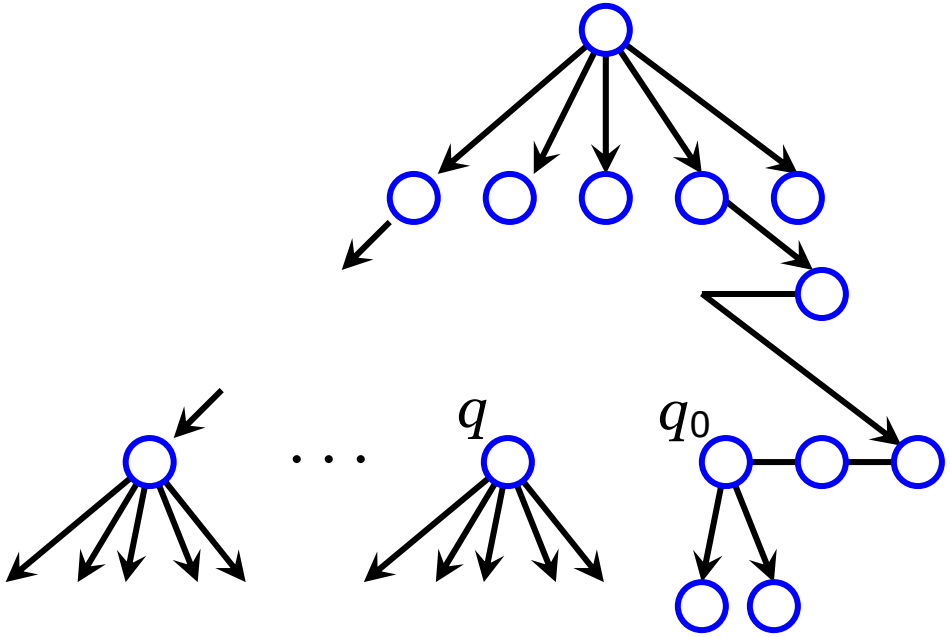
Соответствует ли T оптимальному коду при p_i = 1/8?

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, 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

$$q_0 = \begin{cases} h, & \text{если } h \geq 2, \\ q, & \text{если } h = 1, \\ q-1, & \text{если } h = 0. \end{cases} \qquad (6)$$

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

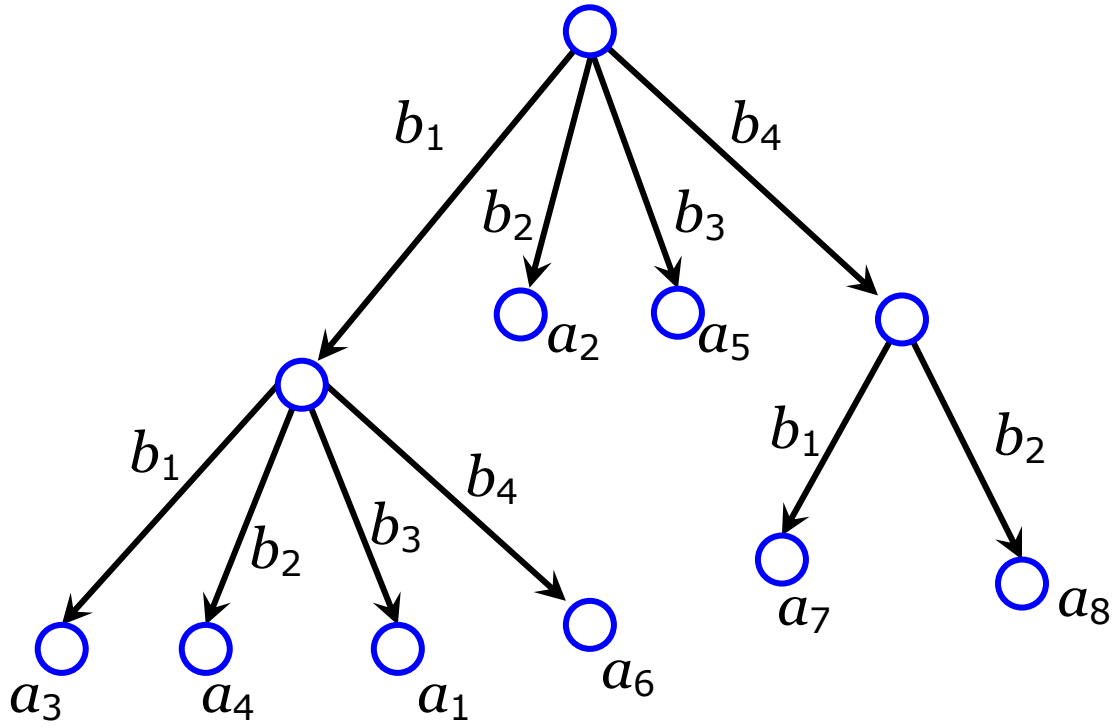
a_4 — b_1b_2 — 0,11

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

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

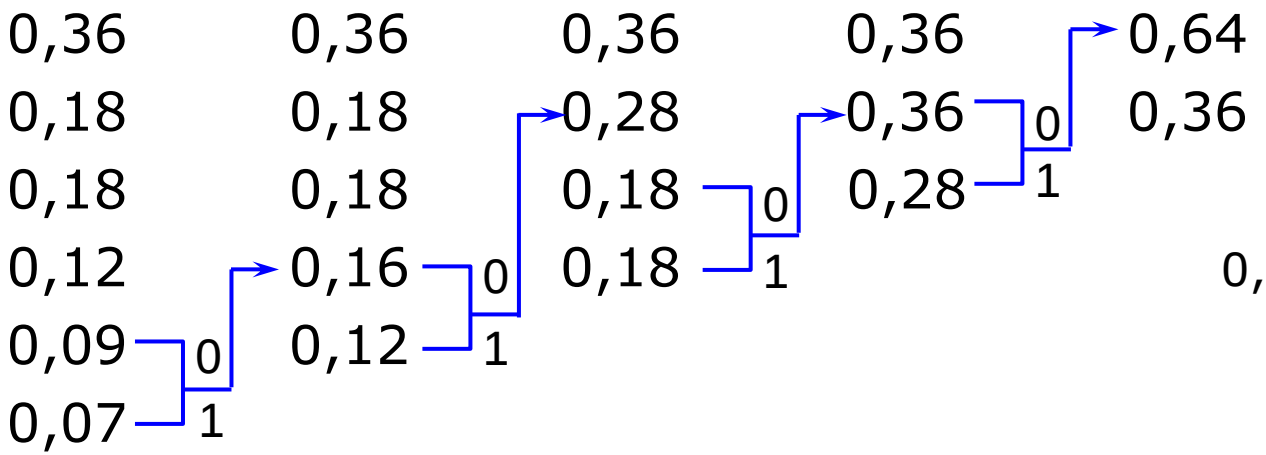
Если бы для алфавита $\mathcal{A}' = \{a_1, \ldots, a_{r-q_0}, a'_{r-q_0+1}\}$ и набора вероятностей $(p_1, \ldots, p_{r-q_0+1})$ нашлась схема Θ' префиксного кодирования с меньшей избыточностью чем $l_{cp}(\Sigma) - p'_{r-q_0+1}$, то подвесив в кодовом дереве для Θ' к вершине v вершины $\{a_r, a_{r-1},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

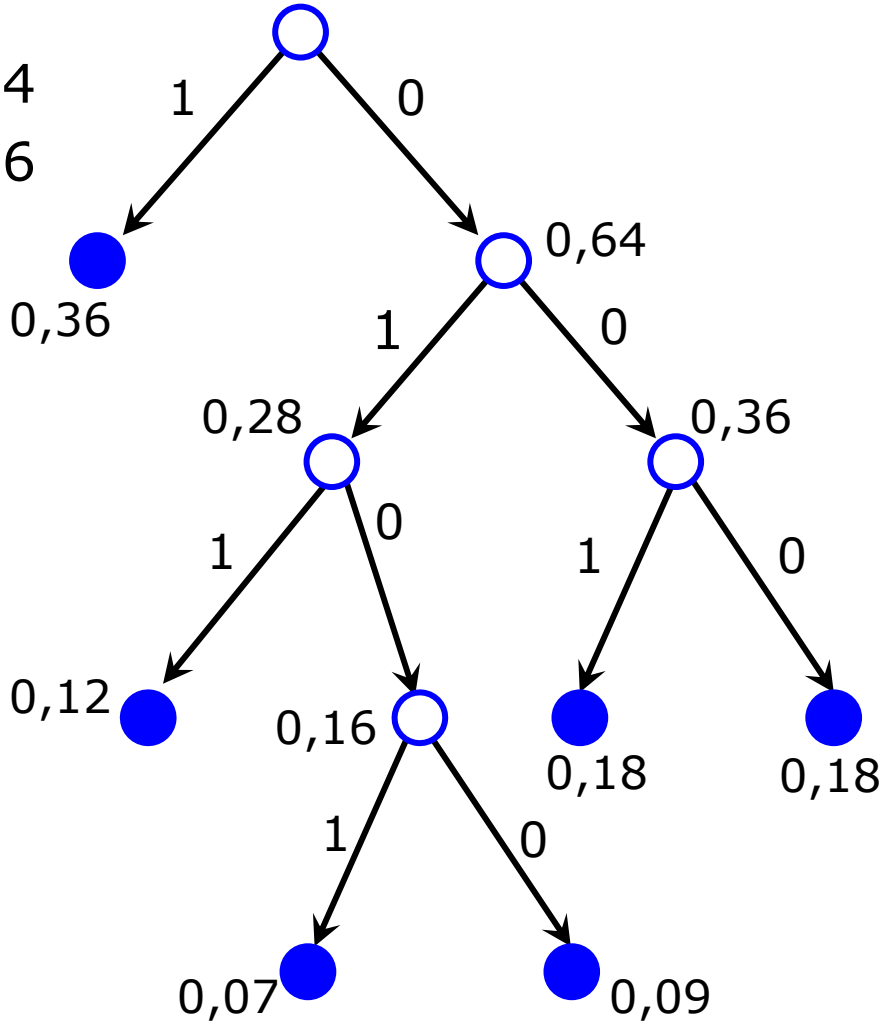
Данная теорема в сочетании с предыдущими леммами дает следующий алгоритм построения кодов с минимальной избыточностью.

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

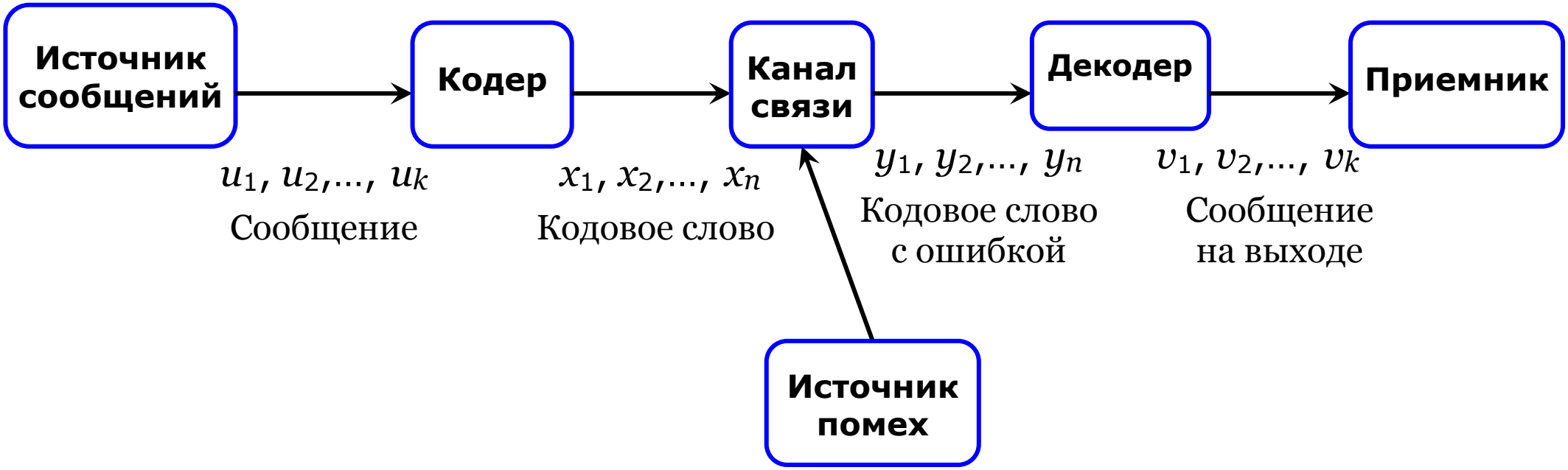
Прямой ход

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 (6), 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.

Кодирование сообщений



Самокорректирующиеся коды

Рассмотрим одну из простейших ситуаций, когда сообщение может искажаться в канале связи. Пусть $\mathcal{A} = \{0, 1\}$ — алфавит, содержащий два символа. Предположим, что в канале связи действует источник помех, который в слове длины l искажает не более p символов. Возникает вопрос: для какого m можно все m-буквенные слова в алфавите $\mathcal{A}$ закодировать l-буквенными словами так, чтобы по коду на выходе закодированные слова однозначно восстанавливались? И как это сделать?

Если $l > 2p$, то $m \geq \lfloor l/(2p + 1) \rfloor$. Действительно, каждую букву можно писать $2p + 1$ раз подряд. Но такое кодирование не является самым экономным. Ниже будет построен *код Хэмминга* для $p = 1$.

Линейный код

$x_1, \ldots, x_k = u_1, \ldots, u_k, \quad x_i \in \{0,1\}$ — символы самого сообщения и $x_{k+1}, \ldots, x_n$ — проверочные символы. Они выбираются так, чтобы кодовое слово удовлетворяло уравнению

$$Hx^T = 0,$$

где H — $((n-k) \times n)$-матрица, называемая *проверочной матрицей кода*.

$H = [\, A \mid I_{n-k} \,]$, I_{n-k} — единичная матрица $\begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix}$

Все операции по модулю 2: 1 + 1 = 0, 1 = – 1.

Пример 4.

Проверочная матрица $H = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & & \\ 1 & 0 & 1 & & 1 & \\ 1 & 1 & 0 & & & 1 \end{array}\right]$

$\begin{cases} 3 \text{ информационные позиции} \\ 3 \text{ проверочные позиции} \end{cases}$

Код с k=3; n=6; Если сообщение $u_1\, u_2\, u_3$, то код $x_1 = u_1$; $x_2 = u_2$; $x_3 = u_3$;

$$\begin{cases} x_2 \oplus x_3 \oplus x_4 = 0 \\ x_1 \oplus x_3 \oplus x_5 = 0 \\ x_1 \oplus x_2 \oplus x_6 = 0 \end{cases}$$

Предположим, что u = (0,11),

тогда x_1=0; x_2=1; x_3=1; x_4=0; x_5=1; x_6= 1;

то есть x = (0,1,1,0,1,1)

Проверка на четность! Сумма должна быть четной.

Т.к. k = 3, то сообщений может быть 2^3 = 8 и всего 8 кодовых слов

Упражнение. Закодировать u = (1 1 0), (1 0 1), (1 1 1).

Пример 5.

Проверочная матрица $\left[\begin{array}{c|cccc} 1 & 1 & & & \\ 1 & & 1 & & \\ 1 & & & 1 & \\ 1 & & & & 1 \end{array}\right]$

Код с повторением. Всего 2 кодовых слова $\begin{cases} 0\ 0\ 0\ 0\ 0 \\ 1\ 1\ 1\ 1\ 1 \end{cases}$

Порождающая матрица

Пусть $H = [\, A \mid E_{n-k} \,]$ — проверочная матрица для некоторого кода, тогда

$G = [E_k \mid -A^T \,]$ — порождающая матрица, то есть любое кодовое слово получается линейной комбинацией строк G по модулю 2.

Пример 6. $H = \left(\begin{array}{cccc|ccc} 1 & 1 & 1 & 0 & 1 & & \\ 1 & 0 & 1 & 1 & & 1 & \\ 0 & 1 & 1 & 1 & & & 1 \end{array}\right)$ Найти порождающую матрицу и выписать все кодовые слова

$$G = \left[\begin{array}{cccc|ccc} 1 & & & & 1 & 1 & 0 \\ & 1 & & & 1 & 0 & 1 \\ & & 1 & & 1 & 1 & 1 \\ & & & 1 & 0 & 1 & 1 \end{array}\right] \Rightarrow \text{Код:} \begin{cases} 1\,1\,0\,0\,0\,1\,1 \\ 1\,0\,1\,0\,0\,0\,1 \\ \ldots\ldots\ldots\ldots \\ 0\,0\,0\,0\,0\,0\,0 \end{cases}$$ 16 вершин.

Линейность кода, вес вектора, кодовое расстояние

Пусть G — порождающая матрица и x, y — кодовые слова, тогда $x \oplus y$ и $x - y$ — тоже кодовые слова, то есть код замкнут относительно операции $\oplus$.

$$w(x) = \sum_{i=1}^{n} x_i$$ — *вес вектора*

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

$$D = \min_{\substack{x,y \in C \\ x \neq y}} \rho(x,y)$$ — *кодовое расстояние*

Кодовое расстояние линейного кода равно минимальному весу его ненулевых векторов:

$$D = \min_{x \neq y} \rho(x,y) = \min_{x \neq y} w(x-y) = \min_{z \in C \;\; z \neq 0} w(z).$$

Упражнение. Посчитать кодовое расстояние для примера 6.

Код Хэмминга

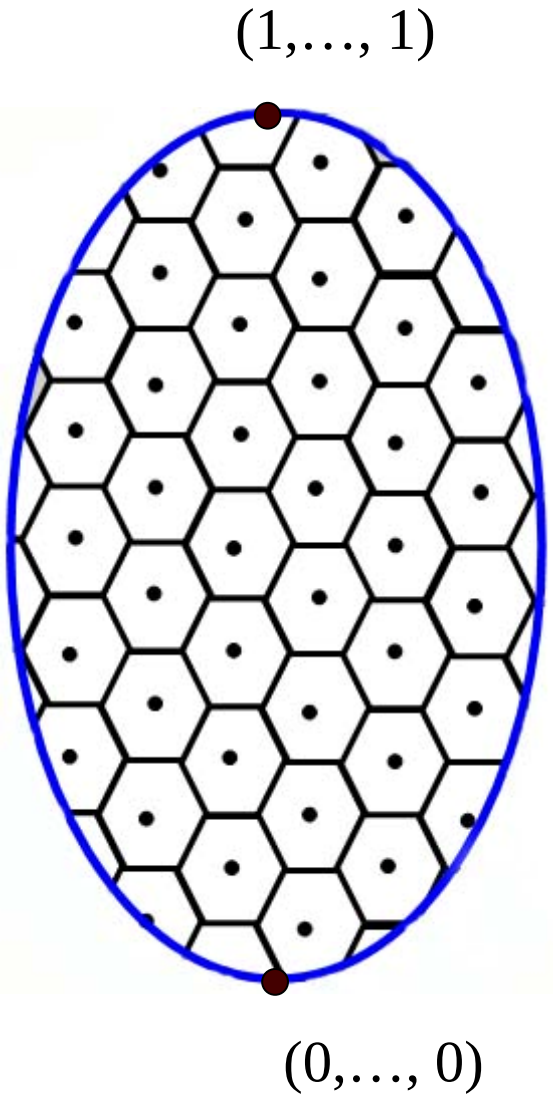
Для любого $r > 0$ код Хэмминга задается проверочной матрицей H_n

$$H = \left(\begin{array}{c|ccc} & 1 & & \\ & & 1 & \\ & & & \ddots \\ & & & 1 \end{array} \right)$$

всего $n = 2^r - 1$ столбцов

Столбцы матрицы — все ненулевые вектора длины r.

Упражнение. Проверить, что код из примера 6 является совершенным, то есть объединение шаров радиуса 1 равно E^n и шары не пересекаются.



Исправление ошибки

Пусть канал связи допускает не более одной ошибки на 7 символов и для кодирования применяется код Хэмминга H_7 (см пример 6).

На приемном конце получим y = (1001110). Это не кодовое слово!

Произвести процедуру декодирования и узнать, что было передано.

По определению $Hx^T = 0$, пусть $y = x + e$, $e = \underbrace{(0 \ldots \overset{i}{1} \ldots 0)}_{7}$, тогда

$Hy^T = H(x+e)^T = He^T$ $\quad Hy^T = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = He^T$ — синдром вектора y, он равен тому

столбцу проверочной матрицы, где была ошибка, то есть 4-й столбец $\Rightarrow$

$\Rightarrow x = y \oplus e$ = (1001110) + (0001000) = (1000110).

Упражнение. Получили y =(1111000), y = (0000001), y = (1111111).

Раскодировать сообщения.