Современная теория информации

Лекция 3. Энтропия на сообщение дискретного источника. Префиксные коды.

Содержание лекции

1. Энтропия на сообщение дискретного источника.
2. Префиксные коды.
3. Неравенство Крафта.
4. Прямая и обратная теоремы побуквенного кодирования.
5. Код Хаффмана.

Энтропия на сообщение

Рассмотрим $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ из $X_1 X_2 \ldots X_n = X^n$.
Энтропия $H(X_1 X_2 \ldots X_n) = H(X^n)$ называется n-**мерной энтропией** процесса.

Энтропия на символ для последовательности длины n определяется как:

$$H_n(X) = \frac{H(X^n)}{n}.$$

Другой способ:

$$H(X_n | X_1, \ldots, X_{n-1}) = H(X | X^{n-1}).$$

Энтропия на сообщение:

$$\lim_{n\to\infty} H_n(X) \text{ и } \lim_{n\to\infty} H(X|X^n).$$

Энтропия на сообщение

Theorem

Для дискретного стационарного процесса (источника)

A. $H(X|X^n)$ *не возрастает с увеличением* n;

B. $H_n(X)$ *не возрастает с увеличением* n;

C. $H_n(X) \geq H(X|X^{n-1})$;

D. $\lim\limits_{n\to\infty} H_n(X) = \lim\limits_{n\to\infty} H(X|X^n)$.

Энтропия на сообщение

Доказательство.

A. Следует из невозрастания энтропии с увеличением числа условий.

C. $H(X^n) = H(X) + H(X|X^1) + ... + H(X|X^{n-1}) \geq$

$$\geq \quad nH(X|X^{n-1}) \geq nH(X|X^n)$$

B. $H(X^{n+1}) \overset{(a)}{=} H(X_1 ... X_n X_{n+1})$

$$\begin{aligned} &\overset{(b)}{=} \quad H(X_1 ... X_n) + H(X_{n+1}|X_1, ..., X_n) \\ &\overset{(c)}{\leq} \quad nH_n(X) + H_n(X) \\ &\overset{(d)}{=} \quad (n+1)H_n(X). \end{aligned}$$

$$H(X^{n+1}) \leq (n+1)H_n(X).$$

$$\Longrightarrow \frac{H(X^{n+1})}{n+1} = H_{n+1}(X) \leq H_n(X)$$

Энтропия на сообщение

Доказательство.

D1. $H_n(X)$ и $H(X|X^n)$ ограничены снизу (≥ 0) и не возрастают, т.е., существуют пределы $\lim\limits_{n\to\infty} H_n(X)$ и $\lim\limits_{n\to\infty} H(X|X^n)$.

Из C следует, что:

$$\lim_{n\to\infty} H_n(X) \geq \lim_{n\to\infty} H(X|X^n).$$

D2. Для $m < n$:

$$\begin{aligned} &H(X^n) = H(X_1 \dots X_n) = \\ &\overset{(a)}{=} H(X_1 \dots X_m) + H(X_{m+1}|X_1, \dots, X_m) + \dots + H(X_n|X_1, \dots, X_{n-1}) \\ &\overset{(b)}{\leq} {}^{a} mH_m(X) + (n-m)H(X|X^m). \end{aligned}$$

После деления на n: $\lim\limits_{n\to\infty} H_n(X) \leq H(X|X^m)$, для любого m.

Устремляем $m \to \infty$:

$$\lim_{n\to\infty} H_n(X) \leq \lim_{m\to\infty} H(X|X^m).$$

□

[a] В правой части учитываем m предыдущих символов, вместо n

Энтропия на сообщение

Обозначим $H_\infty(X) = \lim\limits_{n\to\infty} H_n(X)$, $H(X|X^\infty) = \lim\limits_{n\to\infty} H(X|X^n)$, тогда

$$H_\infty(X) = H(X|X^\infty)$$

Два способа кодирования:

- Расширение алфавита: буквы это последовательности исходных букв длины n.
- Учёт зависимости текущей буквы от n предшествующих букв.

Энтропия на сообщение

Дискретный стационарный источник без памяти

- $H(X_1 ... X_n) = H(X_1) + ... + H(X_n)$.
- $H(X^n) = nH(X)$.
- $H_n(X) = H(X)$,
- $H_\infty(X) = H(X)$.
- $H(X|X^n) = H(X_{n+1}|X_1, ..., X_n) = H(X)$,
- $H(X|X^\infty) = H(X)$.

Отсюда не следует, что для такого источника нужно кодировать каждую букву независимо от других.

Энтропия на сообщение

Марковский источник

- $H(X|X^n) = H(X_{n+1}|X_1, ..., X_n) = H(X_{n+1}|X_{n-s+1}, ..., X_n) = H(X|X^s)$.
- $H(X|X^\infty) = H(X|X^s)$.
- $H(X^n) = H(X_1 \dots X_s X_{s+1} \dots X_n)$

$$= H(X_1 \dots X_s) + H(X_{s+1} \dots X_n|X_1, \dots, X_s).$$

- $H(X_{s+1} \dots X_n|X_1, \dots, X_s) = H(X_{s+1}|X_1, \dots, X_s) +$

$$\begin{aligned} &+ \quad H(X_{s+2}|X_2, \dots, X_{s+1}) + \dots \\ &+ \quad H(X_n|X_{n-s}, \dots, X_{n-1}) \\ &= \quad (n-s)H(X|X^s). \end{aligned}$$

- $\frac{H(X^n)}{n} = \frac{sH_s(X)}{n} + \frac{(n-s)H(X|X^s)}{n}$

Выводы

- Информационная производительность дискретного источника без памяти определяется его энтропией $H(X)$.
- $H(X) \leq \log |X|$, равенство когда все символы равновероятны.
- Энтропия на символ дискретного стационарного источника определяется как $\lim\limits_{n\to\infty} H_n(X) = \lim\limits_{n\to\infty} H(X|X^n)$
- Наилучшее сжатие может быть достигнуто либо кодированием длинных блоков символов, либо с учётом длинной предыстории для каждого символа.

Неравномерное побуквенное кодирование[1]

Рассмотрим дискретный источник без памяти.

- $X = \{1, ..., M\}$, $\{p_1, ..., p_M\}$. $C = \{\boldsymbol{c}_1, ..., \boldsymbol{c}_M\}$, кодовые слова длины $l_1, \ldots, l_M$.
- Средняя длина кодового слова:

$$\bar{l} = \mathsf{E}[l_i] = \sum_{i=1}^{M} p_i l_i$$

$H(X)$ – нижняя граница для $\bar{l}$.

[1] Variable-length coding (VLC)

Неравномерное побуквенное кодирование

Код Морзе

Буква	Кодовое слово
e	$\cdot$
a	$\cdot -$
j	$\cdot - - -$
q	$- - \cdot -$

Декодировать: $\cdot - - - \cdot -$

Неравномерное побуквенное кодирование

Код Морзе

Буква	Кодовое слово
e	$\cdot$
a	$\cdot -$
j	$\cdot - - -$
q	$- - \cdot -$

Декодировать: $\cdot - - - \cdot - \rightarrow$ *aq* или *ja* ?

Для однозначного декодирования кода необходимы разделители (“паузы”) между кодовыми словами.

Префиксные коды

Пример: $X = \{a, b, c, d\}$

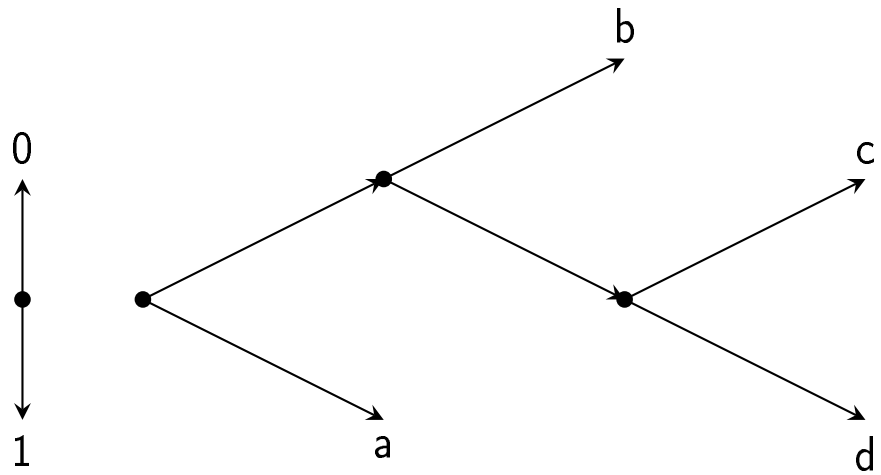


Рис.: Пример двоичного кодового дерева

Буква	Кодовое слово
a	1
b	00
c	010
d	011

Декодировать: 0101001010 ...

Префиксные коды

Свойства:

- Код называется ***префиксным***, если ни одно кодовое слово не является началом другого кодового слова.
- Префиксный код является однозначно декодируемым.
- Если только листья двоичного дерева соответствуют кодовым словам, то код является префиксным.
- Однозначно декодируемый код не обязательно является префиксным.
- Древовидный код является префиксным.

Префиксные коды

Пример

$X = \{0, 1, 2, 3\}$

Какой код является

- префиксным?
- однозначно декодируемым?

1. $C_1 = \{00, 01, 10, 11\}$;
2. $C_2 = \{1, 01, 001, 000\}$;
3. $C_3 = \{1, 10, 100, 000\}$;
4. $C_4 = \{0, 1, 10, 01\}$;
5. $C_5 = \{0, 1, 12, 31\}$;

Неравенство Крафта

Theorem

Необходимым и достаточным условием существования префиксного кода объёмом с длинами кодовых слов $l_1, ..., l_M$ является выполнение неравенства:

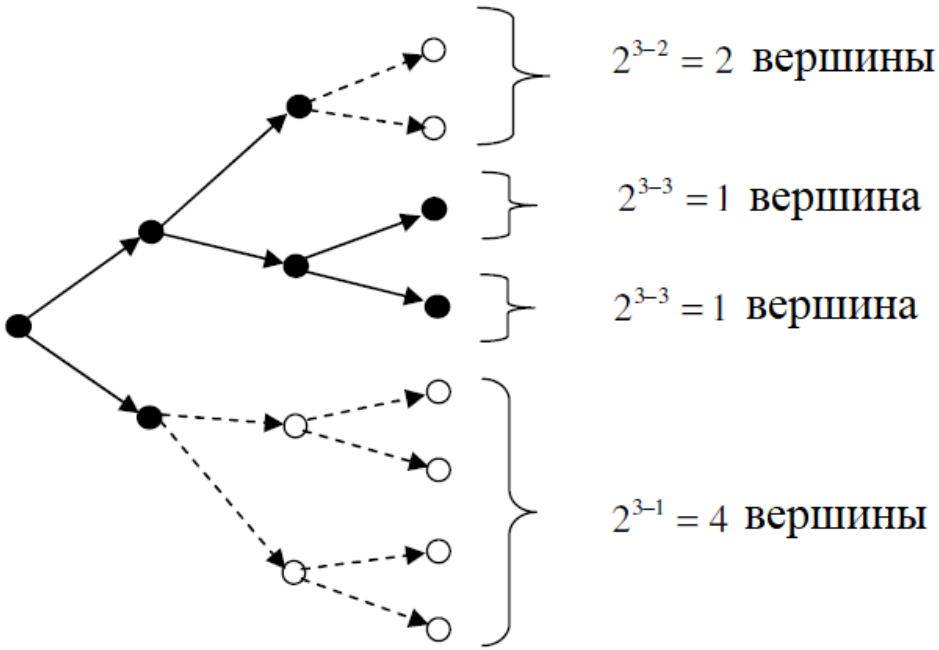
$$\sum_{i=1}^{M} 2^{-l_i} \leq 1.$$

Неравенство Крафта

Необходимость

Неравенство верно для любого префиксного кода.

Выберем L, такое что $L \geq \max_i l_i$. Концевая вершина исходного дерева, расположенная на глубине l_i, имеет 2^{L-l_i} потомков на глубине L.



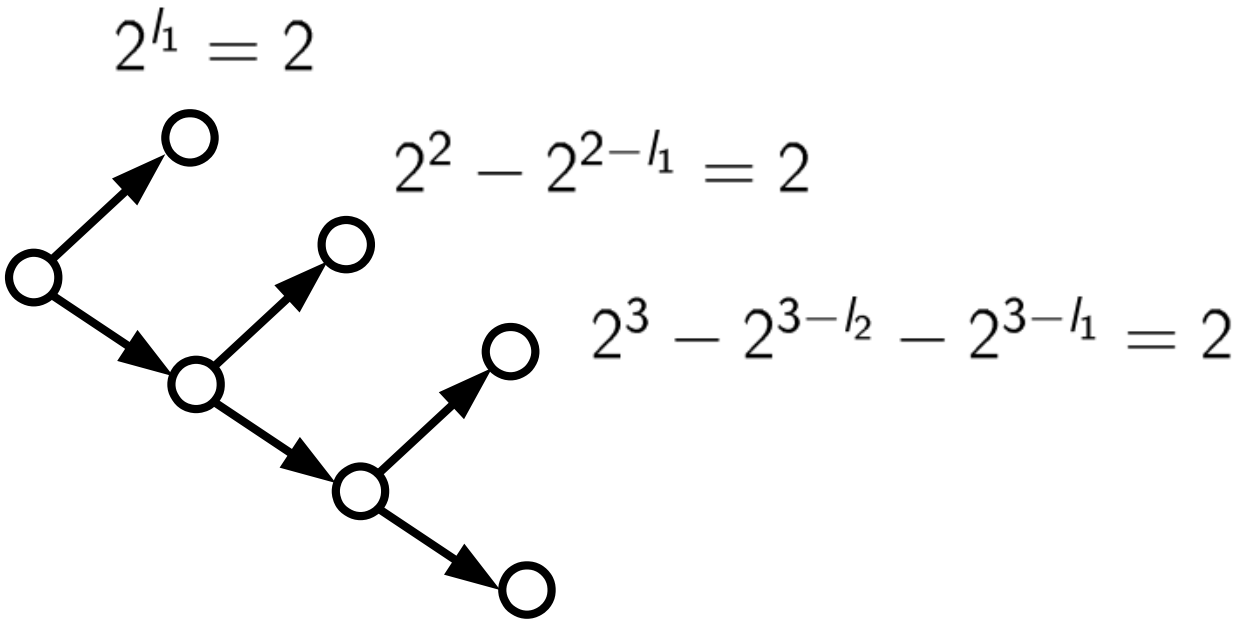
$$\sum_{i=1}^{M} 2^{L-l_i} \leq 2^L.$$

Неравенство Крафта

Достаточность

Если неравенство Крафта выполняется, то существует код с заданным набором длин кодовых слов.

Пример: $l_1 = 1,\ l_2 = 2,\ l_3 = l_4 = 3.$



На каждом шаге количество "свободных" вершин равно 2, т.е., можно поместить следующее кодовое слово.

Неравенство Крафта

Достаточность

Сортируем $\{l_i\}$ по убыванию.

Пошагово строим код:

$$2^{l_2} - 2^{l_2 - l_1} \geq 1$$

$$2^{l_3} - 2^{l_3 - l_2} - 2^{l_3 - l_1} \geq 1$$

...

$$2^{l_M} - 2^{l_M - l_{M-1}} - 2^{l_M - l_{M-2}} - \ldots - 2^{l_M - l_1} \geq 1$$

Однозначно декодируемый код

Theorem

Для любого однозначно декодируемого двоичного кода объёмом M с длинами кодовых слов $l_1, \ldots, l_M$ справедливо неравенство:

$$\sum_{i=1}^{M} 2^{-l_i} \leq 1.$$

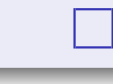
Прямая теорема неравномерного побуквенного кодирования

Theorem

Для ансамбля $X = \{x, p(x)\}$ с энтропией $H(X) = H$ существует побуквенный неравномерный префиксный код со средней длиной кодовых слов $\bar{l} < H + 1$.

Доказательство.

1. Пусть $l_i = \lceil -\log p_i \rceil$. Тогда $\sum\limits_{i=1}^{M} 2^{-l_i} = \sum\limits_{i=1}^{M} 2^{-\lceil -\log p_i \rceil} \leq \sum\limits_{i=1}^{M} 2^{\log p_i} = 1 \Rightarrow$ такой префиксный код существует.
2. $\bar{l} = \sum\limits_{m=1}^{M} p_m l_m < \sum\limits_{m=1}^{M} p_m(-\log p_m + 1) < H + 1.$



Обратная теорема неравномерного побуквенного кодирования

Theorem

Для любого однозначно декодируемого кода для дискретного источника $\{X, p(x)\}$ *с энтропией* H, $\bar{l} \geq H$.

Доказательство.

$$
\begin{aligned}
H - \bar{l} &= -\sum_{x \in X} p(x) \log p(x) - \sum_{x \in X} p(x) l(x) \\
&= \sum_{x \in X} p(x) \log \frac{2^{-l(x)}}{p(x)} \leq \log e \sum_{x \in X} p(x) \left(\frac{2^{-l(x)}}{p(x)} - 1 \right) \\
&\leq \log e \left(1 - \sum_{x \in X} p(x) \right) = 0
\end{aligned}
$$

Оптимальный побуквенный код.

Свойства оптимального кода:

1. Если $p_i < p_j$, то $l_i \geq l_j$.
2. Не менее двух кодовых слов имеют одинаковую длину $l_M = \max\limits_m l_m$. Если у нас имеется только одно кодовое слово максимальной длины, то код не оптимален, так как мы можем убрать последний символ такого кодового слова.
3. Среди кодовых слов длиной $l_M = \max\limits_m l_m$ найдутся два слова, различающиеся только в одном последнем символе.

Оптимальный побуквенный код.

Свойства оптимального кода:

4. Пусть $p_1 \geq p_2 \geq ... \geq p_M$.
   - Для ансамбля $X = \{1, ..., M\}$ и кода C, удовлетворяющего свойствам 1–3, введем ансамбль $X' = \{1, ..., M-1\}$, сообщениям которого приписаны вероятности $\{p'_1, ..., p'_{M-1}\}$ так, что

$$\begin{aligned} p'_1 &= p_1, \\ p'_2 &= p_2, \\ p'_{M-1} &= p_{M-1} + p_M. \end{aligned}$$

   - Из кода C построим код C' для ансамбля X', приписав сообщениям $x'_1, ..., x'_{M-2}$ те же кодовые слова, что и в коде , т.е. $c'_i = c_i$, а сообщению x'_{M-1} слово c'_{M-1}, как общую часть слов C_{M-1} и C_M.
   - Тогда, если C' оптимален для X', то код C оптимален для X.

Оптимальный побуквенный код.

Доказательство свойства 4

Из свойства 3 следует, что:

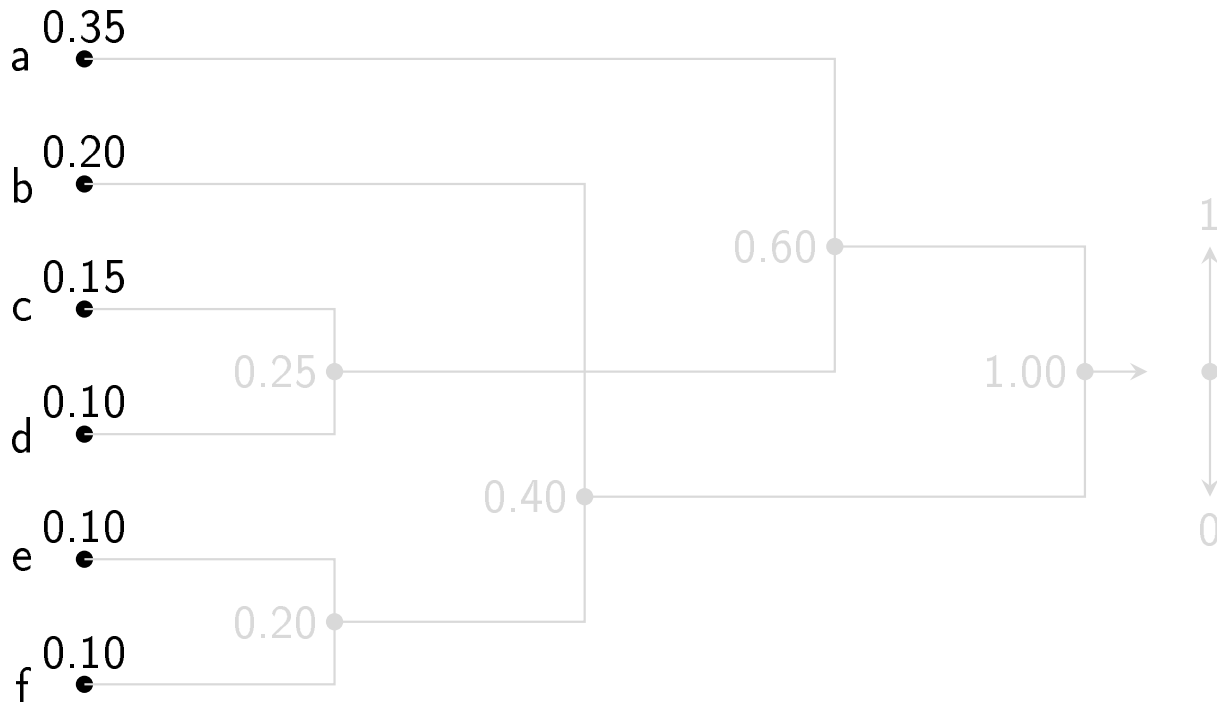
$$l_m = \begin{cases} l'_m \text{ для } m \leq M-2, \\ l'_{M-1}+1 \text{ для } m = M-1, M. \end{cases}$$

Тогда средняя длина кодового слова:

$$\begin{aligned} \bar{l} &= \sum_{m=1}^{M} p_m l_m = \sum_{m=1}^{M-2} p_m l_m + p_{M-1} l_{M-1} + p_M l_M = \\ &= \sum_{m=1}^{M-2} p_m l_m + (p_{M-1}+p_M)(l'_{M-1}+1) = \\ &= \sum_{m=1}^{M-2} p'_m l'_m + p'_{M-1} l'_{M-1} + p_{M-1} + p_M = \\ &= \sum_{m=1}^{M-1} p'_m l'_m + p_{M-1} + p_M = \bar{l}' + p_{M-1} + p_M. \end{aligned}$$

где $\bar{l}' = \sum_{m=1}^{M-1} p'_m l'_m$ – среднее длина кодового слово кода C'.

Код Хаффмана

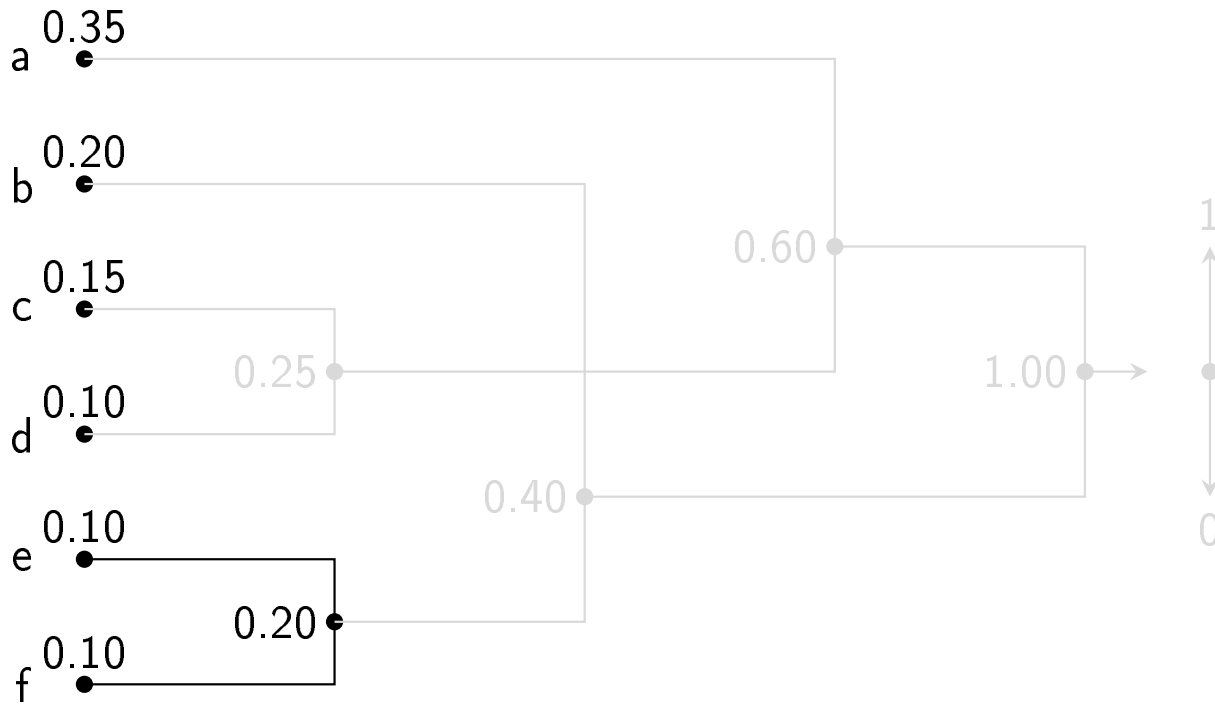


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

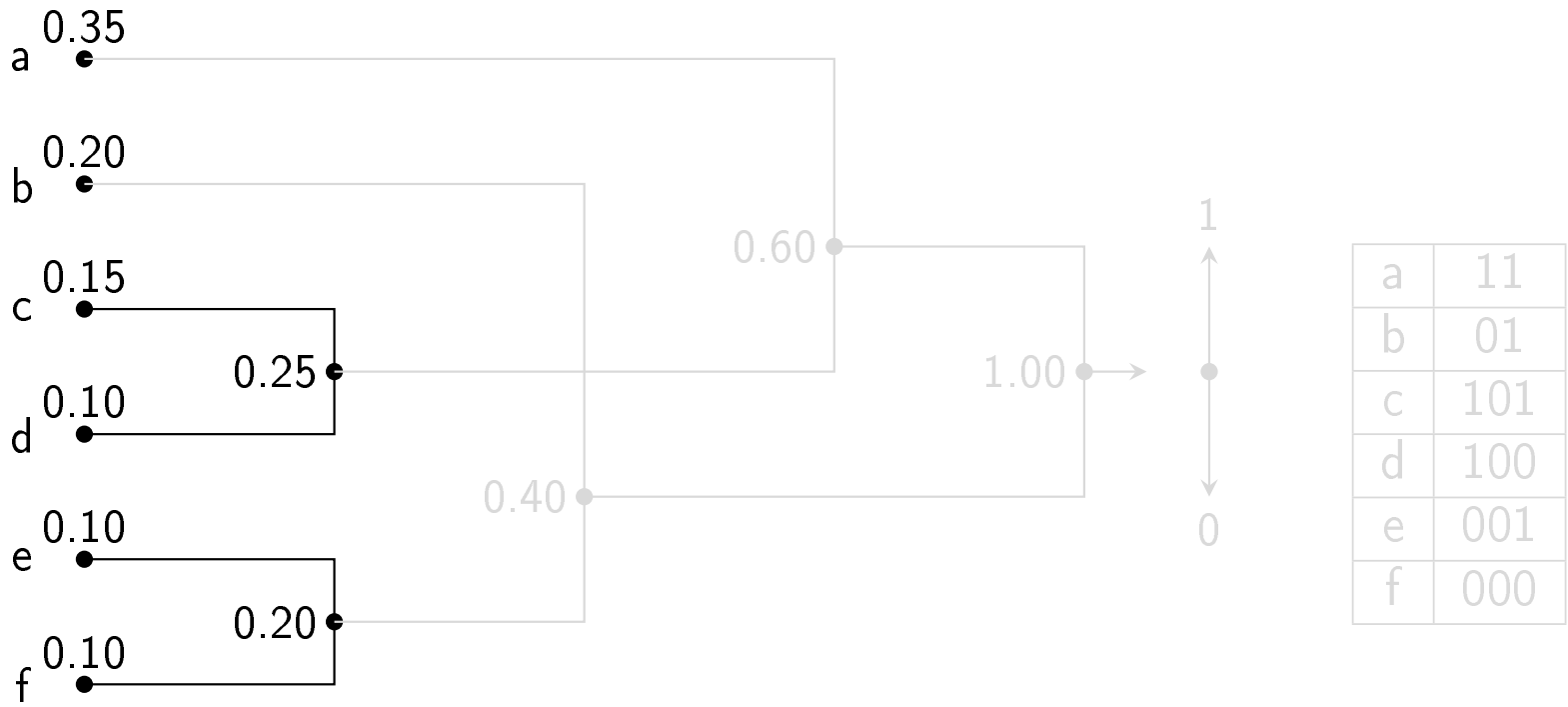


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

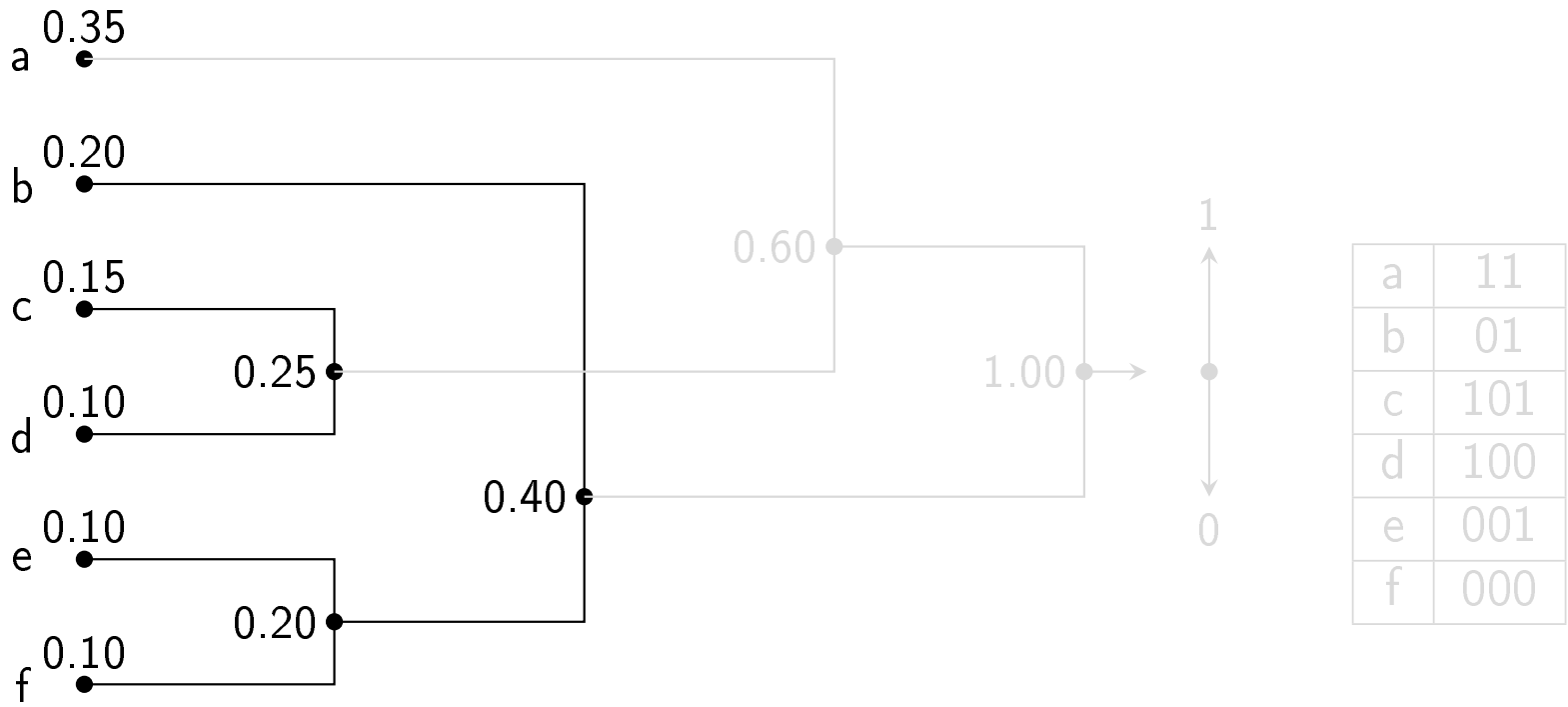


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

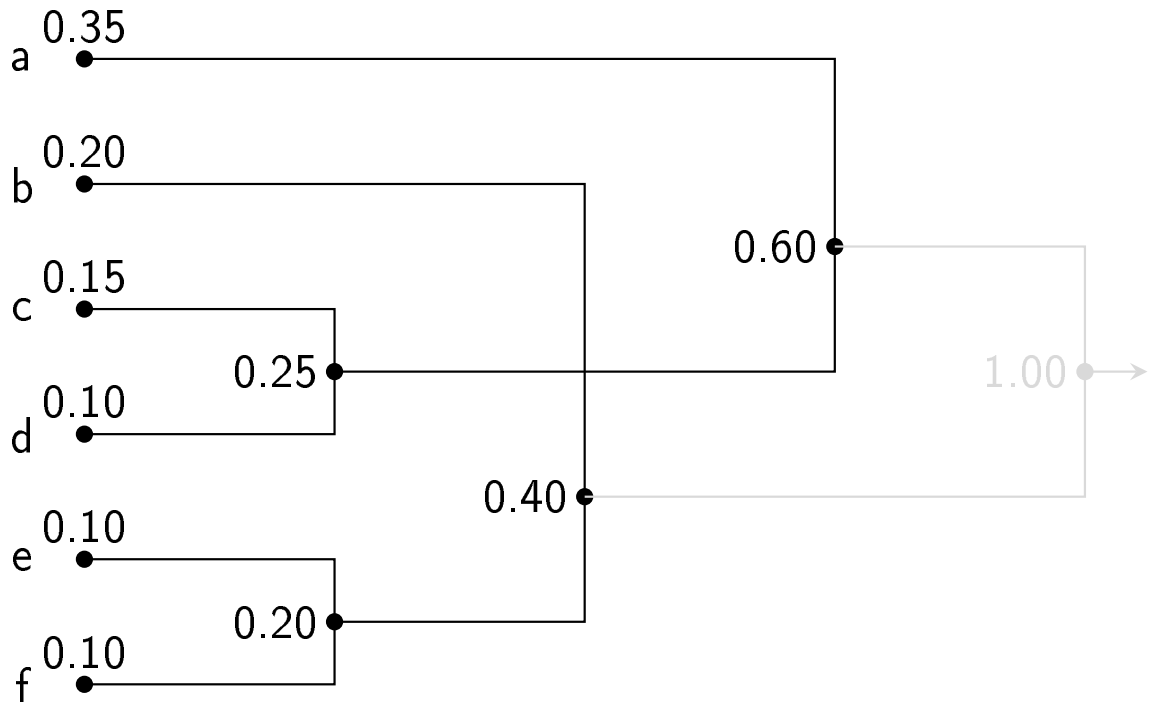


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

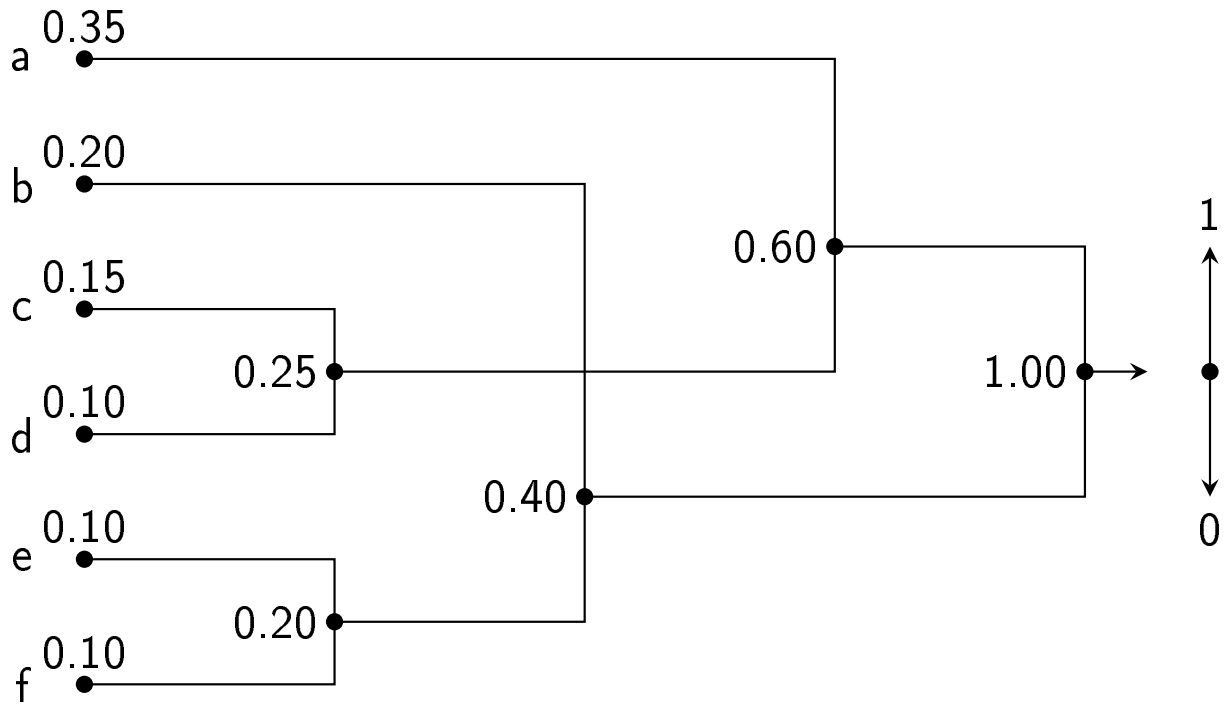


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

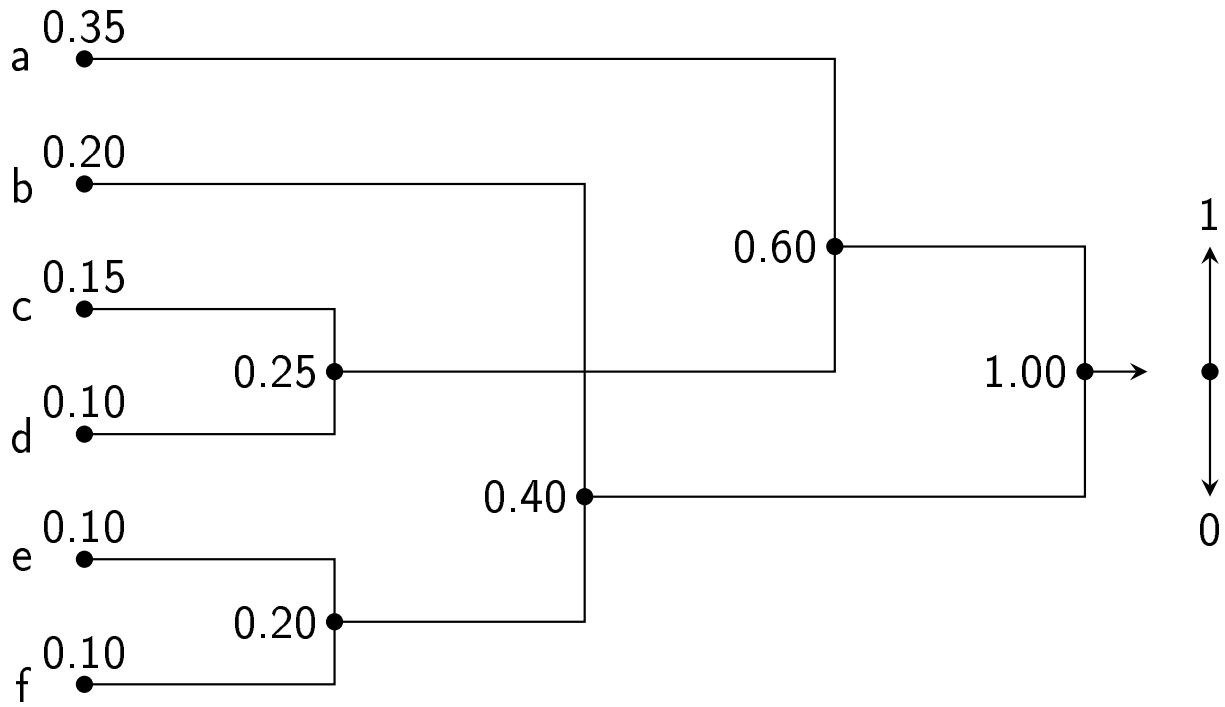


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x) p(x) = 2.4500$$

Код Хаффмана

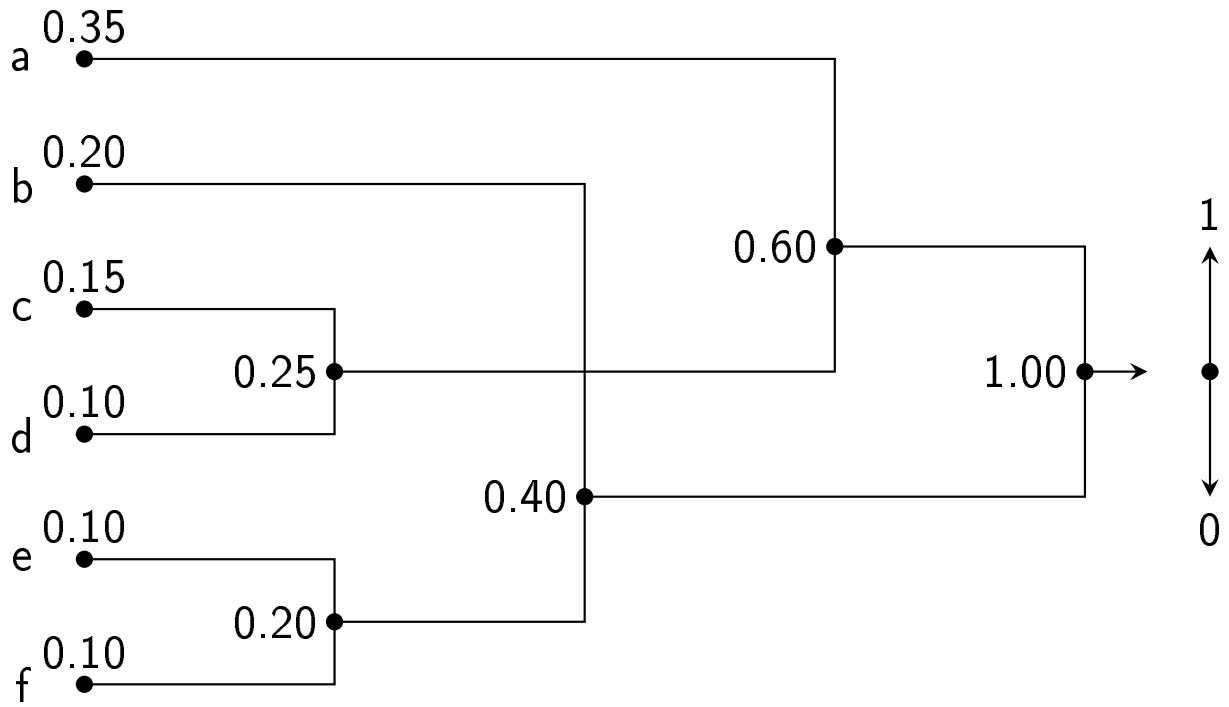


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x)p(x) = 2.4500$$

Код Хаффмана

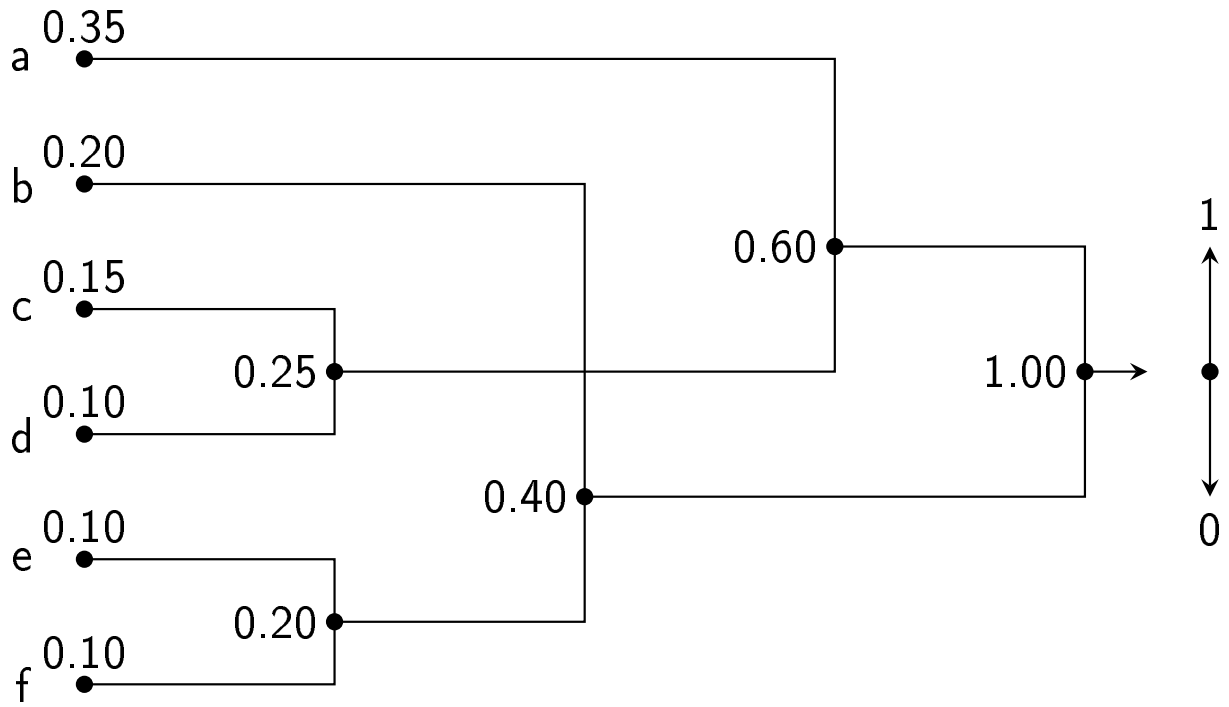


a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x)p(x) = 2.4500$$

Код Хаффмана



a	11
b	01
c	101
d	100
e	001
f	000

$$H = -\sum_x p(x) \log p(x) = 2.4016$$

$$\bar{l} = \sum_x l(x)p(x) = 2.4500$$

Построение кода Хаффмана без обхода дерева

- P – матрица, в которой хранятся вероятности появления сообщений.
- L – матрица длин кодовых слов.
- C – матрица кодовых слов размером $M \times M$.
- T – матрица потомков узлов $M \times M$.

$$P = \begin{pmatrix} 0.5 \\ 0.25 \\ 0.125 \\ 0.125 \end{pmatrix}, C = \begin{pmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{pmatrix}, T = \begin{pmatrix} 0 & - & - & - \\ 1 & - & - & - \\ 2 & - & - & - \\ 3 & - & - & - \end{pmatrix}, L = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Построение кода Хаффмана без обхода дерева

1. Два сообщения x_2 и x_3 с минимальными вероятностями объединяются. Вероятность нового сообщения записывается во вторую строку P.
2. $C[2, L[2]] \leftarrow 0,\ C[3, L[3]] \leftarrow 1.$
3. Во вторую строку матрицы T записываются номера потомков объединенного узла, то есть все номера узлов, которые находились в строках 2 и 3.
4. $L[2] \leftarrow L[2] + 1,\ L[3] \leftarrow L[3] + 1.$

$$P = \begin{pmatrix} 0.5 \\ 0.25 \\ 0.25 \\ - \end{pmatrix}, C = \begin{pmatrix} - & - & - & - \\ - & - & - & - \\ 0 & - & - & - \\ 1 & - & - & - \end{pmatrix}, T = \begin{pmatrix} 0 & - & - & - \\ 1 & - & - & - \\ 2 & 3 & - & - \\ - & - & - & - \end{pmatrix}, L = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}.$$

Построение кода Хаффмана без обхода дерева

- Объединяются x_1 и x_{23}.
- $C[1, L[1]] \leftarrow 0,\ C[2, L[2]] \leftarrow 1,\ C[3, L[3]] \leftarrow 1.$
- В $T[1][..]$ записываются номера потомков, которые находились в первой и второй строках.
- $L[T[1][..]] \leftarrow L[T[1][..]] + 1.$

$$P = \begin{pmatrix} 0.5 \\ 0.5 \\ - \\ - \end{pmatrix}, C = \begin{pmatrix} - & - & - & - \\ 0 & - & - & - \\ 0 & 1 & - & - \\ 1 & 1 & - & - \end{pmatrix}, T = \begin{pmatrix} 0 & - & - & - \\ 1 & 2 & 3 & - \\ - & - & - & - \\ - & - & - & - \end{pmatrix}, L = \begin{pmatrix} 0 \\ 1 \\ 2 \\ 2 \end{pmatrix}.$$

Построение кода Хаффмана без обхода дерева

- Объединяются x_0 и x_{123}.
- $C[0, L[0]] \leftarrow 0$,
  $C[1, L[1]] \leftarrow 1,\ C[2, L[2]] \leftarrow 1,\ C[3, L[3]] \leftarrow 1.$
- В $T[0][..]$ записываются номера потомков, которые находились в нулевой и первой.
- $L[T[0][..]] \leftarrow L[T[0][..]] + 1.$

$$P = \begin{pmatrix} 1.0 \\ - \\ - \\ - \end{pmatrix}, C = \begin{pmatrix} 0 & - & - & - \\ 0 & 1 & - & - \\ 0 & 1 & 1 & - \\ 1 & 1 & 1 & - \end{pmatrix}, T = \begin{pmatrix} 0 & 1 & 2 & 3 \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{pmatrix}, L = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 3 \end{pmatrix}.$$

Код Хаффмана

Реализация декодера

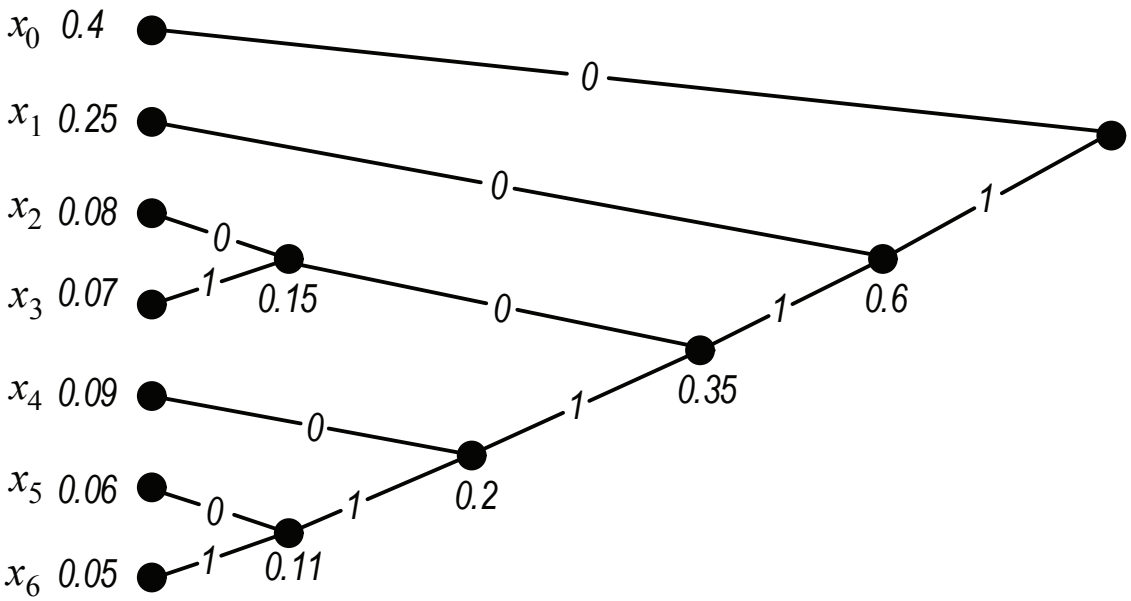
- Так как длина кодовых слов не кратна 8 битам, кодер должен использовать промежуточный буфер в 2-4 байта.
- На практике используется заранее подготовленная таблицу декодирования.

Таблица: Пример однобитной таблицы декодирования для $X = \{x_0, x_1, x_2, x_3\}$ и $C = \{0, 10, 110, 111\}$

Адрес, a	b_j	$f[a]$	Адрес перехода, $A[a]$	$x[a]$
0	0	0	–	x_0
1	1	1	2	–
2	0	0	–	x_1
3	1	1	4	–
4	0	0	–	x_2
5	1	1	–	x_3

Код Хаффмана

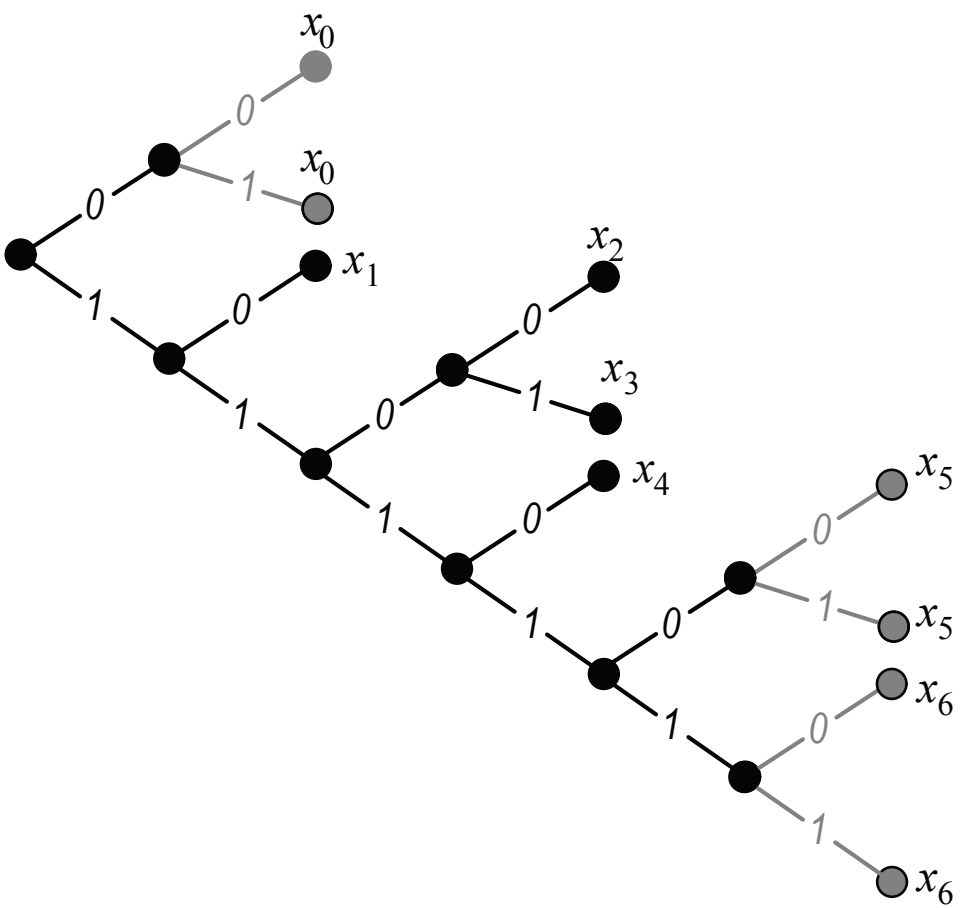
Реализация однобитного декодера



a	b_j	$f[a]$	$A[a]$	$x[a]$
0	0	0	–	x_0
1	1	1	2	–
2	0	0	–	x_1
3	1	1	4	–
4	0	1	6	–
5	1	1	8	–
6	0	0	–	x_2
7	1	0	–	x_3
8	0	0	–	x_4
9	1	1	10	–
10	0	0	–	x_5
11	1	0	–	x_6

Код Хаффмана

Реализация двубитного декодера



a	$b_j b_{j+1}$	$f[a]$	$l[a]$	$A[a]$	$x[a]$
0	00	1	1	–	x_0
1	01	1	1	–	x_0
2	10	1	2	–	x_1
3	11	0	–	4	–
4	00	1	4	–	x_2
5	01	1	4	–	x_3
6	10	1	4	–	x_4
7	11	0	–	8	–
8	00	1	5	–	x_5
9	01	1	5	–	x_5
10	10	1	5	–	x_6
11	11	1	5	–	x_6