

# Кодирование оптимальный код Хаффмана

Лекция 14

# План лекции

- Алфавит, кодирование, код
- Типы кодирования, однозначное декодирование
- Метод кодирования Хаффмана
- Метод кодирования Фано

# Понятие кода

- **Алфавитом** называется конечное множество СИМВОЛОВ
- **Сообщением алфавита  $A$**  называется конечная последовательность символов алфавита  $A$
- Множество всех сообщений алфавита  $A$  обозначается  $A^*$

# Понятие кода

- **Кодом** называется отображение  $K : \text{Алф1}^* \rightarrow \text{Алф2}^*$ , согласованное с конкатенацией, т.е. удовлетворяющее равенству  $K(c_1c_2\dots c_N) = K(c_1) K(c_2)\dots K(c_N)$  для любого сообщения  $c_1c_2\dots c_N$  из  $\text{Алф1}^*$
- Значение  $K(c_1c_2\dots c_N)$  называется **кодом сообщения**  $c_1c_2\dots c_N$
- Код  $K : \text{Алф1}^* \rightarrow \{0,1\}^*$  называется **двоичным кодом**

# Кодирование и декодирование

- **Кодированием сообщения** называется вычисление кода сообщения
- **Декодированием (дешифровкой) сообщения** называется вычисление его прообраза под действием кода
- Код  $K$  называется **однозначно декодируемым**, если существует обратная функция  $K^{-1}$
- Если вычисление  $K^{-1}$  требует большого количества времени, то говорят не о кодировании, а о шифровании

# Пример 1

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 01$ ,  $K(c) = 10$ ,  $K(d) = 1$

$K^{-1}(01101010) = \{\text{addbba, bccc, ...}\}$  – прообраз  
01101010

Данный код не является однозначно  
декодированным

# Пример 2

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 10$ ,  $K(c) = 110$ ,  $K(d) = 111$

Почему данный код является однозначно декодируемым?

# Кодовое дерево

**Кодовым деревом** кода  $K: \text{Алф}1 \rightarrow \text{Алф}2$  называется такое дерево  $T$ , с рёбрами помеченными символами из  $\text{Алф}2$ , что

- Любой путь из корня  $T$  совпадает с началом кода какого-то символа из  $\text{Алф}1$
- Код любого символа из  $\text{Алф}1$  соответствует какому-то пути из корня  $T$ 
  - Почему не всегда до листа?



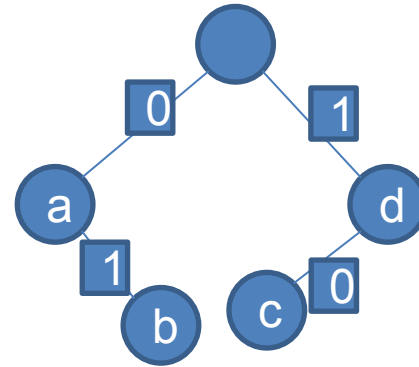
# Пример кодового дерева

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 01$ ,

$K(c) = 10$ ,  $K(d) = 1$



Почему у сообщения 01101010 как минимум два прообраза?

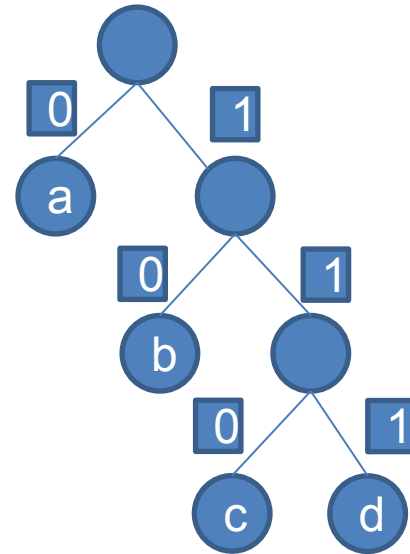
# Пример кодового дерева

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 10$ ,

$K(c) = 110$ ,  $K(d) = 111$



Почему у *любого* сообщения один прообраз?

# Префиксный код

Код  $K$  называется **префиксным**, если для любых двух сообщений  $U$  и  $V$  код  $K(U)$  не является началом (префиксом) кода  $K(V)$  и наоборот

- Свойства префиксного кода
- В дереве префиксного кода коды всех символов заканчиваются в листьях
- Префиксный код позволяет выделять коды символов без использования разделителей

# Примеры префиксных кодов

Пример 1

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 00$ ,  $K(b) = 01$ ,  $K(c) = 10$ ,  $K(d) = 11$

Как выглядит кодовое дерево этого кода?

# Примеры префиксных кодов

Пример 2

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 10$ ,  $K(c) = 110$ ,  $K(d) = 111$

Как выглядит кодовое дерево этого кода?

# Однозначная декодируемость префиксного кода

Теорема Любой префиксный код однозначно декодируем

Доказательство

- Пусть  $K$  – префиксный код. Докажем, что у кода  $S=K(R)$  любого сообщения  $R$  ровно один прообраз
- Индукция по длине  $L$  сообщений  $R$
- База  $L = 1$ 
  - $R$  восстанавливается однозначно в силу префиксности  $K$ 
    - Что было бы, если бы коды *двух разных* символов являлись бы префиксом  $S$
- Шаг  $L > 1$ 
  - $K$  согласован с конкатенацией  $\implies$  найдётся символ  $c$  такой, что  $S = K(c) S'$ 
    - Что бы было бы, если бы такого символа не было бы или бы он был бы не один бы?
  - $K$  префиксный  $\implies$  символ  $c$  единственный
  - Длина прообраза  $S'$  строго меньше длины прообраза  $S$
  - По предположению индукции  $S'$  декодируется однозначно

# Пример

Алф1 = {a,b,c,d}

Алф2 = {0,1}

$K(a) = 0$ ,  $K(b) = 101$ ,  $K(c) = 110$ ,  $K(d) = 1110$

Рассмотрим сообщение 01101010

01101010 =  $K(a)$  1101010

1101010 =  $K(c)$  1010

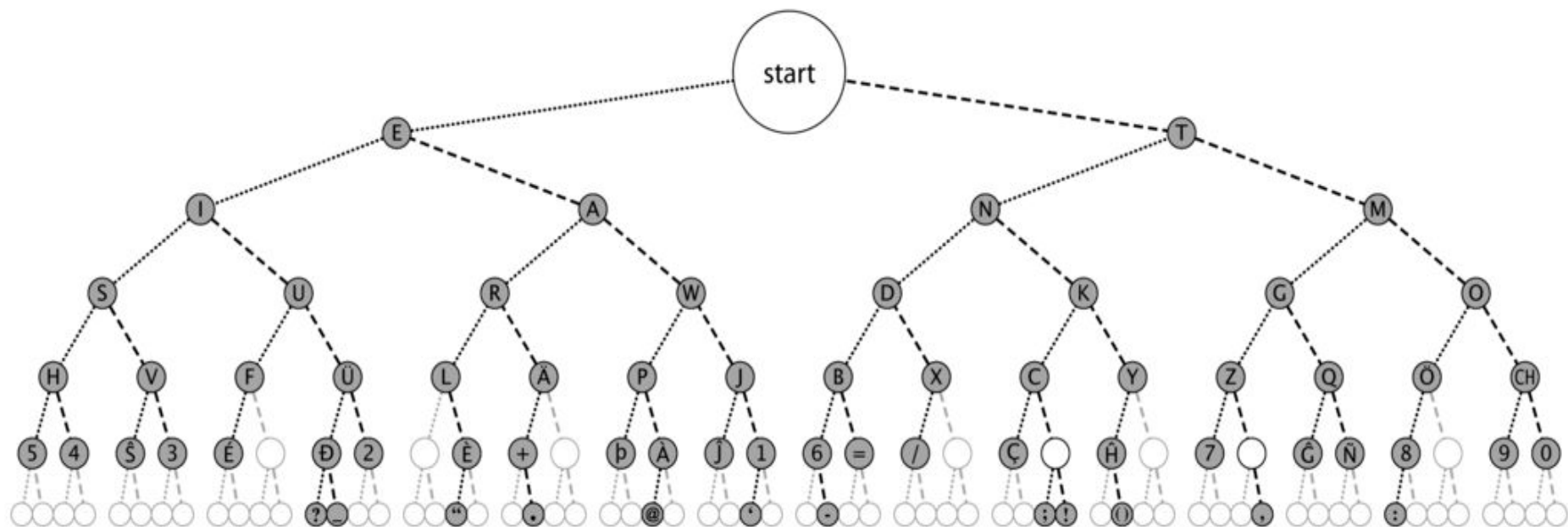
1010 =  $K(b)$  0

0 =  $K(a)$

$K(acba) = 01101010$

# Пример азбука Морзе

- 1840 Alfred Vail по заказу телеграфной компании Samuel F.B. Morse
- Двоичный (точка, тире) непrefixный код – почему?
- Троичный (точка, тире, пауза) префиксный код – почему?
- Кодовое дерево азбуки Морзе как двоичного кода для



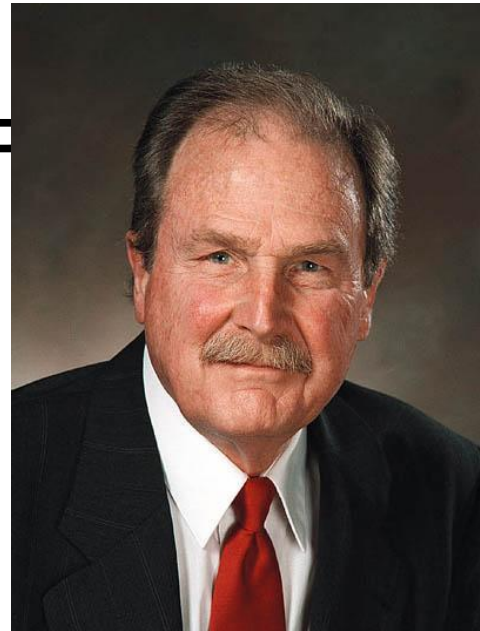


# Понятие оптимального кода

- Обозначим
  - $\Delta$  – множество кодов Алф1\*  $\rightarrow$  Алф2\*
  - $K$  – какой-то код из  $\Delta$
  - $R$  – произвольное сообщение из Алф1\*
  - $L(K, R)$  – длина  $R$  после кодирования
  - $p_x$  – число вхождений символа  $c_x$  в  $R$ 
    - заодно мы пронумеровали символы из Алф1,  $x$  – номер символа  $c_x$
- Длина кода сообщения  $R$  есть  $L(K, R) = \sum p_x \cdot L(K, c_x)$
- Код  $K^*$  называется **оптимальным** для сообщения  $R$  в множестве кодов  $\Delta$ , если
$$L(K^*, R) = \min \{ \text{длина}(K, R) \mid K \in \Delta \}$$

# Оптимальный двоичный префиксный код

- Как *быстро* построить оптимальный двоичный префиксный код для данного сообщения?
- Сжатие данных при хранении и передаче
- Устранение избыточности при шифровании данных
- David A. Huffman 1925-1999 "A Method for the Construction of Minimum-Redundancy Codes", Proceedings of the I.R.E., September 1952, pp 1098–1102.



# Свойства оптимального двоичного префиксного кода

Пусть  $R$  -- сообщение в алфавите  $\text{Алф1}=\{c_1, \dots, c_n\}$

$c_x$  входит в  $R$   $p_x$  раз ( $x=1, \dots, n$ )

$K^*$  -- оптимальный двоичный префиксный код для  $R$

1. Если  $p_x < p_y$ , то  $L_x(K^*) \geq L_y(K^*)$   
– Иначе для кода  $K(c_x) = K^*(c_y)$ ,  $K(c_y) = K^*(c_x)$  и  $K(c) = K^*(c)$   
 $L(K, R) < L(K^*, R)$
2. Можно занумеровать символы  $\text{Алф1}$  так, чтобы  
 $p_1 \geq p_2 \geq \dots \geq p_n$  и  $L(K^*, c_1) \leq L(K^*, c_2) \leq \dots \leq L(K^*, c_n)$

# Свойства оптимального двоичного префиксного кода

3. Символов с кодом длины  $L(K^*, c_n)$  (с самым длинным кодом) не менее двух
  - Иначе удалим последний символ в коде  $c_n$  -- длина  $L(K^*, R)$  сократится, префиксность  $K^*$  сохранится
  
4. Можно перенумеровать символы так, что  $K^*(c_n) = P 0$  и  $K^*(c_{n-1}) = P 1$  и сохранив условие 2
  - Следует из свойства 3

# Свойства оптимального двоичного префиксного кода

5. Оптимальный двоичный префиксный код  $k^*$  для сообщения  $r$ , полученного из сообщения  $R$  заменой самого редкого символа  $c_n$  на  $c_{n-1}$ , и  $K^*$  связаны соотношениями
- $k^*(c_{n-1}) = \text{удалить из } K^*(c_{n-1}) \text{ последний символ}$
  - $K^*(c_n) = k^*(c_{n-1}) 0$
  - $K^*(c_{n-1}) = k^*(c_{n-1}) 1$
  - $K^*(c) = k^*(c)$  для остальных символов  $c$
  - $L(K^*, R) = L(k^*, r) + p_n + p_{n-1}$

# Построение дерева оптимального префиксного двоичного кода

Вход

Кратности  $p_1, \dots, p_n$  вхождений символов  $c_1, \dots, c_n$  в сообщение

Выход

Дерево оптимального двоичного префиксного кода для сообщения

Алгоритм

- $W = \{p_1(c_1), \dots, p_n(c_n)\}$  – множество деревьев
  - Левая скобочная запись, кратности в качестве меток вершин
- пока в  $W$  два или более поддеревьев
  - Найти в  $W$  деревья  $T = x(\dots)$  и  $U = y(\dots)$  с минимальными метками  $x$  и  $y$
  - $W = (W \setminus \{T, U\}) \cup \{(x+y)(T, U)\}$

# Пример

кол около колокола

о – 7; к – 4; л – 4; пробел – 2; а – 1.

Один из вариантов работы алгоритма

Множество  $W$

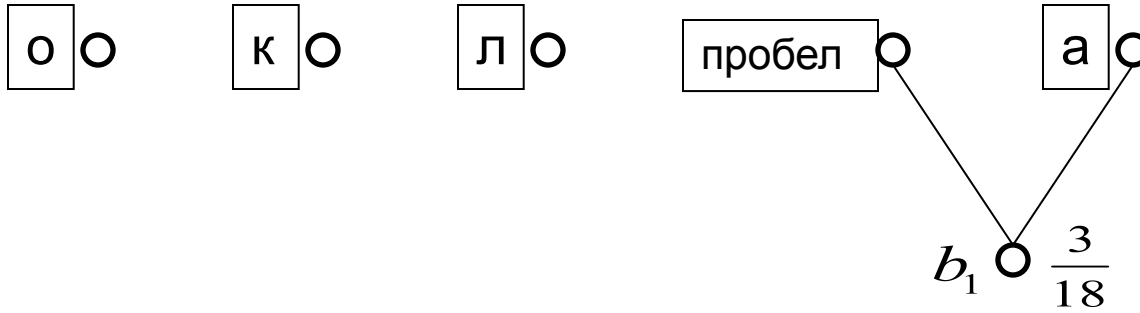
До цикла {7(о), 4(к), 4(л), 2(пробел), 1(а) }

После шага 1 {7(о), 4(к), 4(л), 3(2(пробел), 1(а)) }

После шага 2 {7(о), 4(к), 7(4(л), 3(2(пробел), 1(а))) }

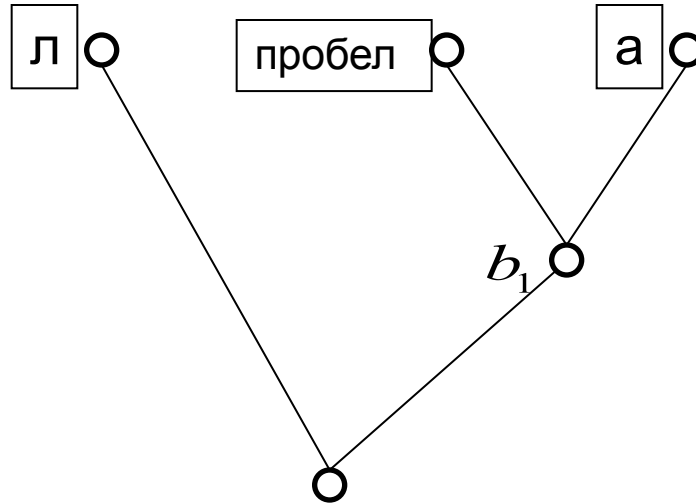
После шага 3 {7(о), 11(4(к), 7(4(л), 3(2(пробел), 1(а)))) }

После шага 4 {18(7(о), 11(4(к), 7(4(л), 3(2(пробел), 1(а)))))) }



Дерево после шага

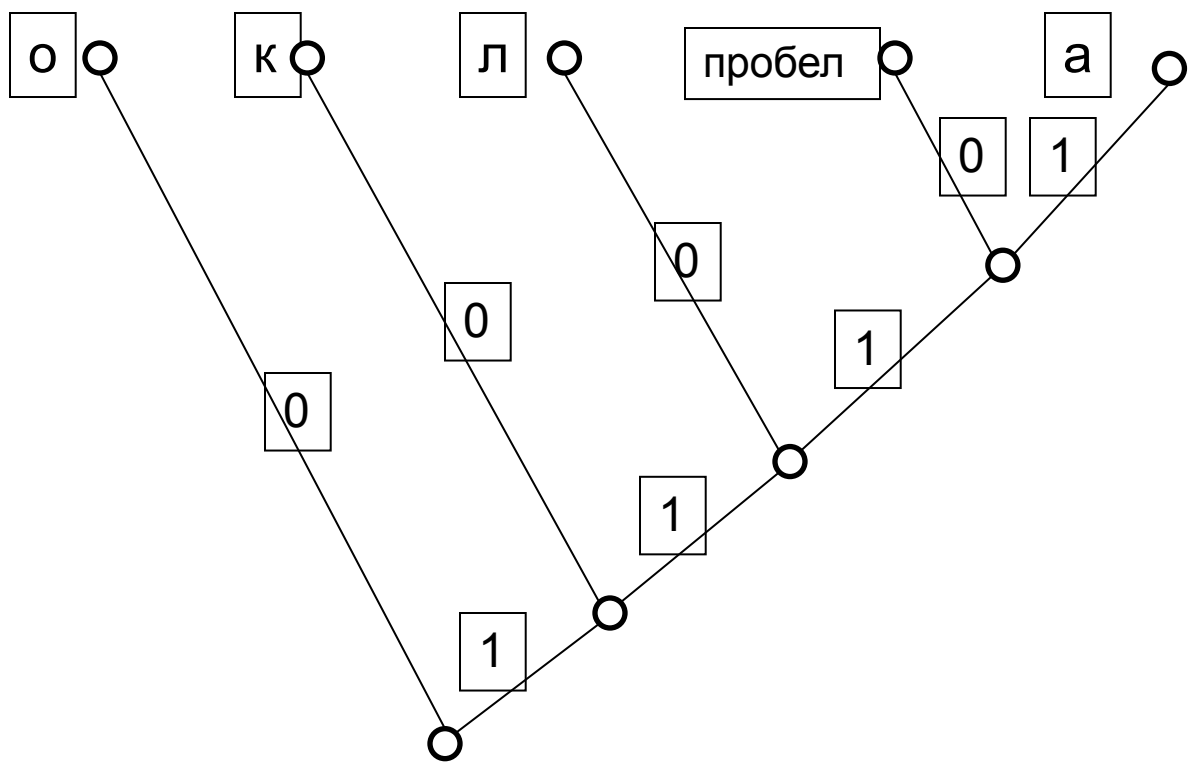
1



Дерево после шага

2





Дерево после шага 4

# Пример построения кода по кодовому дереву

- Пометим дуги, исходящие из каждой вершины дерева, единицей и нулем
- Проходя путь из корня дерева до символа и выписывая все пометки дуг на этом пути, получим код для этого символа

В нашем примере коды будут такими

о	0,	
к	10 пробел	1110
л	110а	1111

Закодированное сообщение

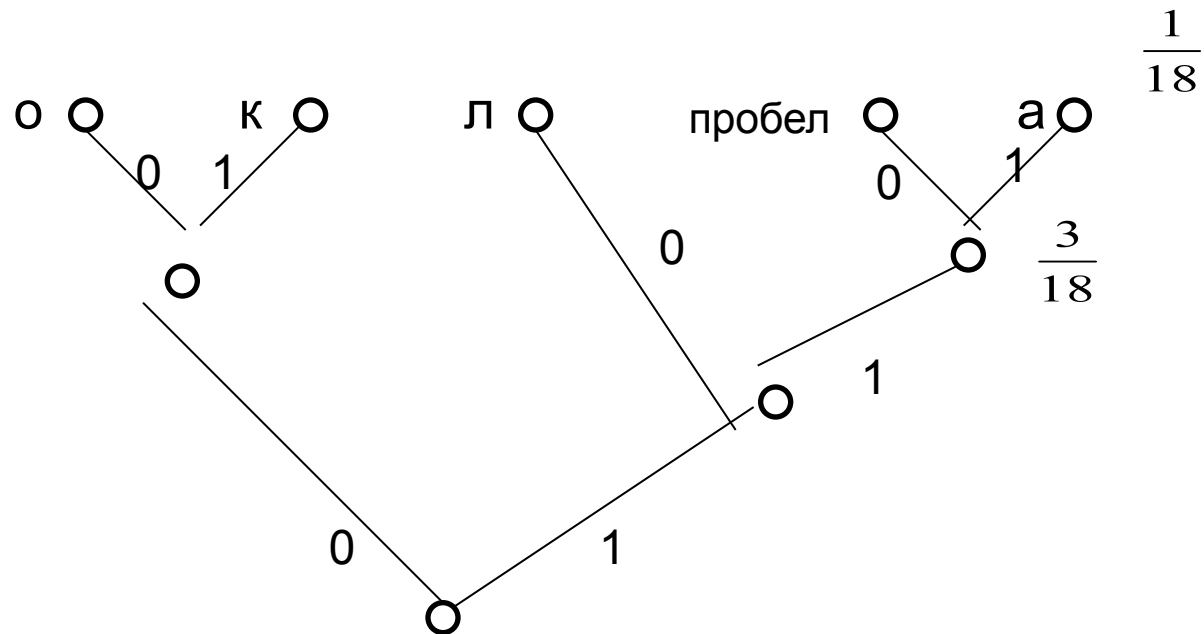
10011011100100110011101001001001101111

Длина закодированного сообщения  $L = 39$

Для разобранного примера можно построить другое дерево

Закодированное сообщение длины  $L = 39$

010010110000100100011001001000010010111



## Теорема

Длина кодового слова в оптимальном префиксном двоичном коде ограничена порядковым номером минимального числа Фибоначчи, превосходящего длину входного текста.

Доказательство – в качестве упражнения

## Следствие

При кодировании по алгоритму Хаффмана текстов ASCII размером до 11Тб код любого символа короче 64 битов

- Алгоритм кодирования код
- Алгоритм декодирования
- Кодирование однозначное декодирование
- Кодирование Фано

- **Метод кодирования Фано**

# Метод Фано

Роберт Марио Фано р. 1917

Один из первых алгоритмов сжатия на основе префиксного кода



# Метод Фано

- Упорядочим входной алфавит по возрастанию частот  $p_1 \leq p_2 \leq \dots \leq p_n$  вхождения символов в сообщение
- Обозначим  $S_k = p_1 + p_2 + \dots + p_k$ ,  $S_0 = 0$
- Строим таблицу  $K$  с двоичными кодами символов входного алфавита
- $K[i][1] = i$ -й символ (по возрастанию частот)
- $K[i][2] = S_k$
- Остальные клетки – на след. слайде

# Метод Фано

- $K[i][j]$  заполняем 0 и 1 по след. правилу
- Для каждого *максимального* интервала строк  $[a, b]$ , у которых в столбце  $j-1$  находятся одинаковые цифры
  - Находим  $c \in [a, b]$  такое, что  $S_c$  ближе всего к  $(S_a+S_b)/2$
  - $K[i][j] = 1$  для  $i \in [a, c]$ ,  $K[i][j] = 0$  для  $i \in [c+1, b]$



# Пример

$A = \{a, b, c, d, e\}$

Частоты  $p_a = 0.11$ ,  $p_b = 0.15$ ,  $p_c = 0.20$ ,  $p_d = 0.24$ ,  $p_e = 0.30$

0.46 ближе к 0.5

0.26 ближе всех к  $(0.00+0.46)/2=0.23$

0.70 ближе всех к  $(0.46+1.00)/2=0.73$

0.11 ближе всех к  $(0.00+0.26)/2=0.13$

	$P_i$	$S_i$			
		0			
a	0.11	0.11	1	1	1
b	0.15	0.26	1	1	0
c	0.20	0.46	1	0	
d	0.24	0.70	0	1	
e	0.30	1.00	0	0	

# Свойства кода Фано

- Кодовое дерево для кода Фано обладает следующим свойством
  - Ребра, исходящие из корня, соответствуют разбиению алфавита на две группы символов, близкие по частоте
  - Ребра, исходящие из вершины следующего «этажа», соответствуют разбиению соответствующей группы на близкие по частоте подгруппы и т. д.
- Код Фано – префиксный код
  - Почему?

# Свойства кода Фано

- Код Фано неоптимальный
- Пример
  - Частоты  $p_1=0.4$ ,  $p_2=p_3=p_4=p_5=0.15$
  - Фано: 00 01 10 110 111
    - средняя длина кодового слова  $2*0.4+(2+2)*0.15+(3+3)*0.15 = 2.3$
  - Хаффман: 0 010 011 000 001
    - средняя длина кодового слова  $1*0.4+(3+3+3+3)*0.15 = 2.2$
  - Как выглядят кодовые деревья кода Хаффмана и Фано?

# Метод Шеннона

- Клод Шеннон 1916 – 2001, основоположник теории информации
1. Упорядочим входные символы по возрастанию частот и образуем частичные суммы  $S_k$  как в методе Фано
  2. Для каждой частоты  $S_k$  находим  $n_k$  т.ч.  $1/2^{n_k} \leq S_k \leq 2/2^{n_k}$  --- нужно отделить одну  $S_k$  от другой
  3.  $S_k$  разлагаем в двочную дробь  $0.d_1d_2d_3\dots$
  4. Первые  $n_k$  цифр этой дроби задают код для  $k$ -го символа

# Пример построения кода Шеннона

	nk	разложение $S_k$	код
$p(a) = 0.08$	$S_a = 0.08$	4 0.0001	0001
$p(b) = 0.12$	$S_b = 0.20$	4 0.0011	0011
$p(c) = 0.15$	$S_c = 0.35$	3 0.010	010
$p(d) = 0.28$	$S_d = 0.63$	2 0.10	10
$p(e) = 0.37$	$S_d = 1.00$	2 0.11	11

Пример вычисления на:

$$0.08 \approx 1/12; \quad 1/2^4 \leq 1/12 \leq 2/2^4$$

# Свойства кода Шеннона

- Код Шеннона -- префиксный код
  - Почему?
- Пусть  $p_k$  – частота вхождения  $k$ -го символа в кодируемое сообщение длины  $N$ .  
Кодирование такого сообщения кодом Шеннона дает сообщение длины не более  $N * (p_1 * \log_2(p_1) + p_2 * \log_2(p_2) + \dots + p_n * \log_2(p_n))$ 
  - Почему? Как Шеннон выбрал длины кодовых слов?

# Заключение

- Алфавит, кодирование, код
- Типы кодирования, однозначное декодирование
- Метод кодирования Хаффмана
- Метод кодирования Фано