СЕВАСТОПОЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

Лекция № 4

«Методы кодирования»

Код с проверкой на четность. Итеративный код. Код с удвоением элементов. Инверсный код. Код Шеннона-Фано. Код Хаффмена.

Ведущий преподаватель: канд. техн. наук, доцент кафедры ИУТС Альчаков Василий Викторович

Код с проверкой на четность

Алгоритм кодирования.

1. Выполняется суммирование по модулю 2 разрядов, образующих код.
2. Вместе с информационной частью кода передается один контрольный разряд. Его значений «0» или «1» выбирается с условием, чтобы сумма цифр в предаваемом коде была равна 0 по модулю 2 (для случая четности) или 1 (для случая нечетности). Допускается что может возникнуть только одна ошибка.

Алгоритм декодирования.

1. Вычисляется сумма по модулю 2 всех разрядов с учетом контрольного.
2. Если результат проверки равен 0, ошибок в кодовой комбинации не найдено. В противном случае – в коде присутствует ошибка.

ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Код с проверкой на четность

– **Исходная кодовая комбинация**

1	0	1	0	1	0	1	1		???

– **Вычисление контрольного разряда**

$$1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = \boxed{1}$$

1	0	1	0	1	0	1	1		1

– **Отсутствие ошибки**

1	0	1	0	1	0	1	1		1

– **Проверка принятой комбинации**

$$1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0$$

Рез-т проверки = 0 – в кодовой комбинации отсутствует ошибка

4 ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Код с проверкой на четность

– **Ошибка кратности 1**

1	0	1	**1**	1	0	1	1		1

– **Проверка принятой комбинации**

$$1 \oplus 0 \oplus 1 \oplus \boxed{1} \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

Рез-т проверки $\neq 0$ – в кодовой комбинации присутствует ошибка

– **Ошибка кратности 2**

1	0	1	**1**	1	**1**	1	1		1

– **Проверка принятой комбинации**

$$1 \oplus 0 \oplus 1 \oplus \boxed{1} \oplus 1 \oplus \boxed{1} \oplus 1 \oplus 1 \oplus 1 = 0$$

Рез-т проверки $= 0$ – в кодовой комбинации отсутствует ошибка

Ошибка не обнаружена!

ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Код с проверкой на четность

– **Ошибка кратности 3**

1	0	0	1	1	1	1	1		1

– **Проверка принятой комбинации**

$$1 \oplus 0 \oplus \boxed{0} \oplus \boxed{1} \oplus 1 \oplus \boxed{1} \oplus 1 \oplus 1 \oplus 1 = 1$$

Рез-т проверки $\neq 0$ – в кодовой комбинации присутствует ошибка

Выводы:

1. Код способен выполнять обнаружение ошибок нечетной кратности.
2. Избыточность кода минимальна.
3. Помехоустойчивые характеристики минимальны.

Итеративный код

Алгоритм кодирования.

1. Исходная кодовая комбинация разбивается на отдельные блоки равной длины. При невозможности сформировать такие блоки допускается дополнить недостающие разряды нулями.
2. Полученные блоки помещаются в матрицу (как правило используется квадратная матрица).
3. Для каждого блока вычисляется контрольный разряд по правилу суммирования по модулю 2. Суммирование выполняется по строкам и столбцам. Полученные кодовые разряды помещаются в конце строки или столбца соответственно.

Алгоритм декодирования.

1. Полученная кодовая комбинация помещается в матрицу.
2. Выполняется проверка по строками и столбцам, аналогичная кодированию.
3. Если проверка не выполняется, строка или столбец помечаются.
4. Искаженные разряды находятся на пересечении помеченных строк и столбцов.

Итеративный код

Общее уравнение кодирования $b = \oplus \sum_{i=1}^{k} a_i$

Исходная комбинация $a_1, a_2, \ldots, a_9$

Разбивка на блоки и запись элементов в матрицу

$$\begin{vmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \\ & & \end{vmatrix} \quad \text{или} \quad \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ & & \end{vmatrix}$$

Итеративный код

Уравнение кодирования по строкам $$b_{rj} = \oplus \sum_{j=1}^{3} a_{ij}; \; i = 1 \div 3$$

Уравнение кодирования по столбцам $$b_{ci} = \oplus \sum_{i=1}^{3} a_{ij}; \; j = 1 \div 3$$

Результирующая матрица
$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & b_{r1} \\ a_{21} & a_{22} & a_{23} & b_{r2} \\ a_{31} & a_{32} & a_{33} & b_{r3} \\ b_{c1} & b_{c2} & b_{c3} & b \end{vmatrix} \qquad b = \oplus \sum_{i=1}^{3} \sum_{j=1}^{3} a_{ij}$$

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9

$\downarrow$

a_1	a_2	a_3	b_{r1}	a_4	a_5	a_6	b_{r2}	a_7	a_8	a_9	b_{r3}	b_{c1}	b_{c2}	b_{c3}	b

Итеративный код

Пример.

Исходная комбинация

1	0	1	0	0	1	1	1	0

Матрица итеративного кода

1	0	1	$1\oplus0\oplus1=\mathbf{0}$
0	0	1	$0\oplus0\oplus1=\mathbf{1}$
1	1	0	$1\oplus1\oplus0=\mathbf{0}$
$1\oplus0\oplus1=\mathbf{0}$	$0\oplus0\oplus1=\mathbf{1}$	$1\oplus1\oplus0=\mathbf{0}$	$1\oplus0\oplus1\oplus0\oplus0\oplus1\oplus1\oplus1\oplus0=\mathbf{0}$

Результирующая комбинация

101**0** 001**1** 110**0** **0100**

ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Итеративный код

Искажение

$\downarrow$

$1 \quad \boxed{1} \quad 1 \quad \mathbf{0} \quad 0 \quad 0 \quad 1 \quad \mathbf{1} \quad 1 \quad 1 \quad 0 \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0}$

1	$\boxed{1}$	1	**0**
0	0	1	**1**
1	1	0	**0**
0	**1**	**0**	**0**

Проверка для 1-й строки: $1 \oplus 1 \oplus 1 = 1 \neq \mathbf{0}$.

Проверка для 2-го столбца: $1 \oplus 0 \oplus 1 = 0 \neq \mathbf{1}$.

1	$\boxed{1}$	1	**0**
0	0	1	**1**
1	1	0	**0**
0	**1**	**0**	**0**

Итеративный код

Выводы:

1. Код способен выполнять обнаружение и исправление ошибок кратности 1 – 100% случаев.
2. Ошибки кратности 2 могут быть обнаружены в 100% случаев.
3. Код обладает большой избыточностью.
4. Помехоустойчивые характеристики высокие.

Код с удвоением элементов

Метод кодирования с удвоением элементов характеризуется наличием дополнительного символа для каждого информационного символа передаваемого кода.

Пример:

1	0	1	1

-- Исходная комбинация

1	0	0	1	1	0	1	0
1	0	1	1	1	0	1	0

-- Закодированная комбинация

-- Комбинация с ошибкой

$1 \rightarrow 10$

$0 \rightarrow 01$

Основное правило кодирования

Показателем искажения являются сочетания типа 00 или 11 в парных элементах. Код не способен исправлять ошибки, приводящие к двукратным противоположным изменениям разрядов в парных элементах. Помехоустойчивость кода выше, чем кода с проверкой на четность, но существенно возрастает избыточность.

Код с удвоением элементов

Характеристики кода:

1. Код не исправляет ошибки, приводящие к двукратным противоположным изменениям разрядов в парных элементах
2. Помехоустойчивость данного кода выше, чем кода с проверкой на четность.
3. Избыточность кода равна 0,5.

$$r = \frac{p}{n}$$

n – значность кода

p – число проверочных символов

Инверсный код

В основе метода лежит повторение кодовой комбинации.

Если исходная комбинация содержит **четное число единиц**, то повторная комбинации **в точности повторяет исходную**. **Иначе** повторение происходит **в инверсном виде**.

$$01010 \rightarrow 01010\underline{01010}$$

$$11010 \rightarrow 11010\underline{00101}$$

Проверка производится суммированием единиц основной комбинации. Если их число четно, то элементы второй части принимаются в том же виде, после чего обе части комбинации сравниваются поэлементно – первый элемент первой части с первым элементом второй части. При несовпадении хотя бы одного элемента принятая комбинация считается неверной.

Инверсный код позволяет обнаружить практически все ошибки приема-передачи. Не обнаруживается лишь одновременное искажение парных элементов в обеих частях кодовой комбинации.

ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Инверсный код

Пример.

Исходная комбинация

0	1	0	1	0

Число 1 четное $\Rightarrow$

0	1	0	1	0	0	1	0	1	0

Прием комбинации

0	1	0	1	0
0	1	0	1	0

Ошибки нет

Прием комбинации

0	1	1	1	0
0	1	0	1	0

Найдено несовпадение

Прием комбинации

0	1	1	1	0
0	1	1	1	0

Ошибка не выявлена

ОСНОВЫ ТЕОРИИ КОДИРОВАНИЯ

Код Шеннона-Фано

Алгоритм кодирования.

1. Все символы алфавита упорядочиваются в порядке убывания вероятности их появления.
2. Кодируемые символы делятся на две равновероятные или приблизительно равновероятные подгруппы.
3. Каждому символу из верхней подгруппы присваивается код «0», а каждому символу из нижней подгруппы – код «1».
4. Каждая из подгрупп снова делится на две равновероятные или приблизительно равновероятные подгруппы. При этом каждому символу из верхней подгруппы присваивается код «0», а из нижней – «1».
5. Деление на подгруппы проводится до тех пор, пока в подгруппе не останется по одному символу.
6. Результирующие кодовые слова записываются слева направо по кодам подгрупп, соответствующих кодируемому символу. Суммарная вероятность появления символов алфавита должна равняться 1.

Код Шеннона-Фано

Пример.

Символ алфавита a_i	Вероятность $P(a_i)$	Шаги алгоритма					Количество элементарных символов l_i	Кодовое слово
		1	2	3	4	5		
a_1	0,30	0 $\boxed{0,55}$	0 $\boxed{0,3}$				2	00
a_2	0,25	0	1 $\boxed{0,25}$				2	01
a_3	0,15	1	0 $\boxed{0,25}$	0 $\boxed{0,15}$			3	100
a_4	0,1	1	0	1 $\boxed{0,1}$			3	101
a_5	0,1	1 $\boxed{0,45}$	1 $\boxed{0,2}$	0 $\boxed{0,1}$			3	110
a_6	0,05	1	1	1 $\boxed{0,1}$	0 $\boxed{0,05}$		4	1110
a_7	0,04	1	1	1	1 $\boxed{0,05}$	0 $\boxed{0,04}$	5	11110
a_8	0,01	1	1	1	1	1 $\boxed{0,01}$	5	11111

Код Хаффмана

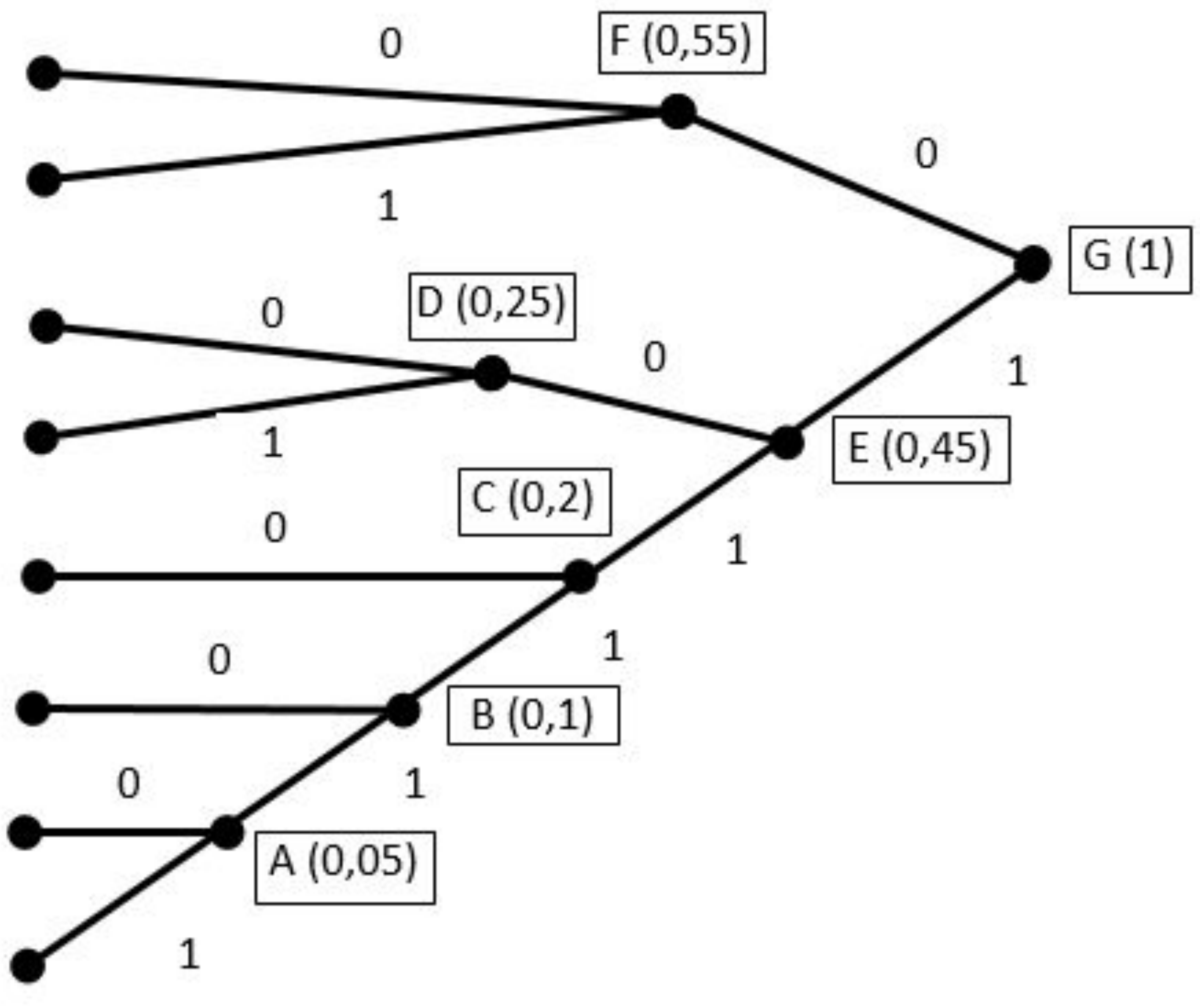
Алгоритм кодирования.

1. Все символы алфавита упорядочиваются в порядке убывания их вероятностей появления.
2. Проводится «укрупнение» символов. Для этого два последних символа «укрупняются» в некоторый вспомогательный символ с вероятностью, которая равняется сумме вероятностей символов, которые были «укрупнены».
3. Образовавшаяся новая последовательность вновь сортируется в порядку убывания вероятностей с учетом вновь образованного за счет «укрупнения» символа.
4. Процедура повторяется до тех пор, пока не получится один «укрупненный» символ, вероятность которого равна 1.

Замечание. На практике не используют многократное выписывание символов алфавита и их упорядочивание.

Код Хаффмана

Пример.

Символ алфавита a_i	Вероятность $P(a_i)$	Кодовое дерево	Количество элементарных символов l_i	Кодовое слово
a_1	0,30		2	00
a_2	0,25		2	01
a_3	0,15		3	100
a_4	0,1		3	101
a_5	0,1		3	110
a_6	0,05		4	1110
a_7	0,04		5	11110
a_8	0,01		5	11111