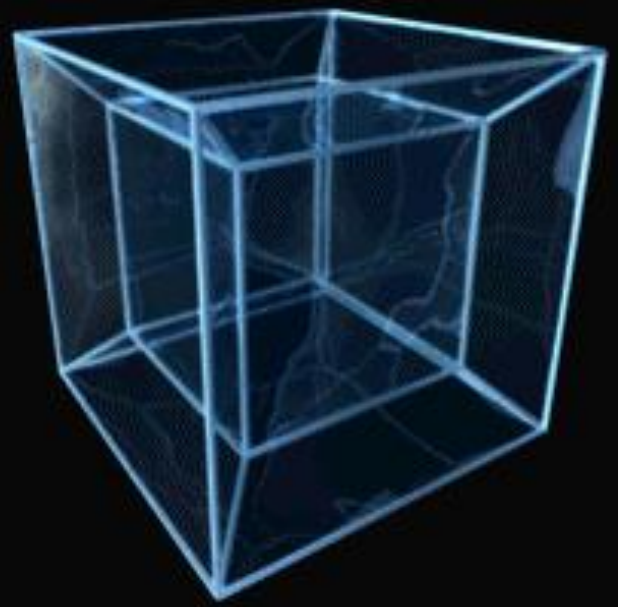
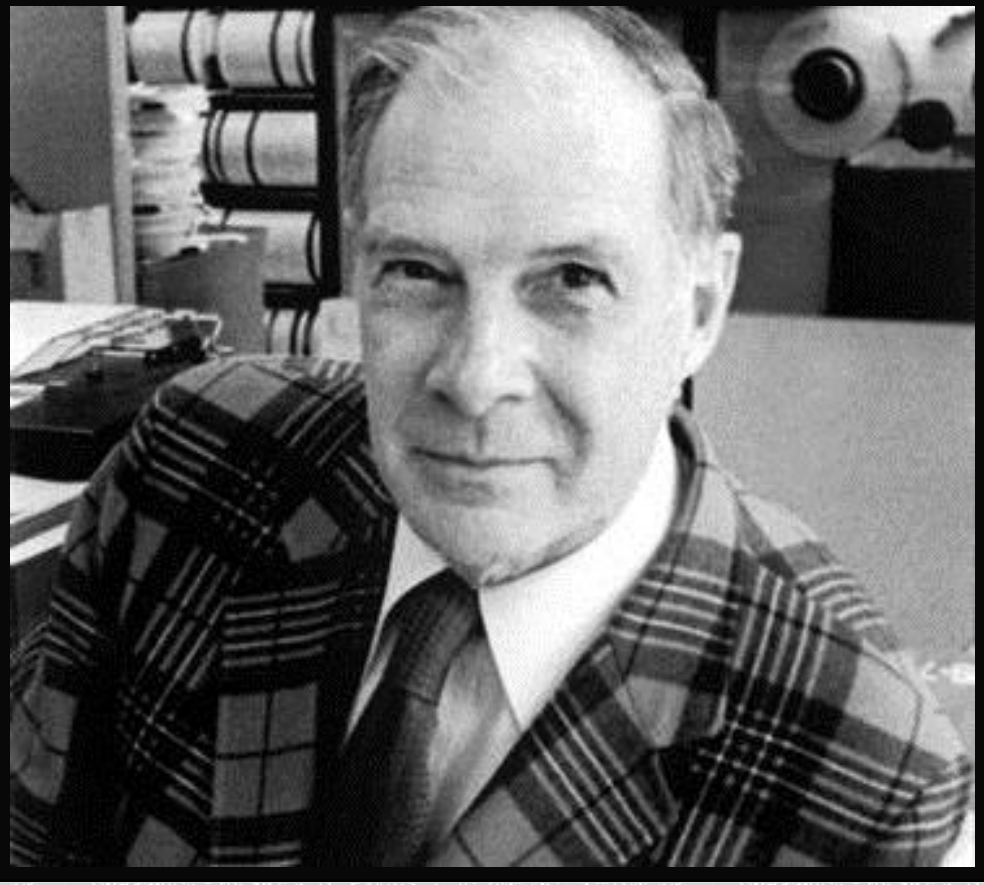




ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

Лекция 8

12 октября 2018 г.

Q/A

Регистрация

piazza.com/tversu.ru/other/cs101

Основная сраница

piazza.com/tversu.ru/other/cs101/home

ПОМЕХОУСТОЙЧИВОЕ КОДИРОВАНИЕ

ОПРЕДЕЛЕНИЯ

Опр. 1: Помехоустойчивые коды – это такие коды, которые позволяют обнаружить ошибку на принимающей стороне.

Опр. 2: Корректирующие коды – помехоустойчивые коды, которые позволяют исправить фиксированное количество ошибок.

Опр. 3: Блоковые коды – разбивают сообщение на блоки, каждый из которых кодируется независимо.

Опр. 4: Информационные символы – символы кода, совпадающие с символами исходного сообщения.

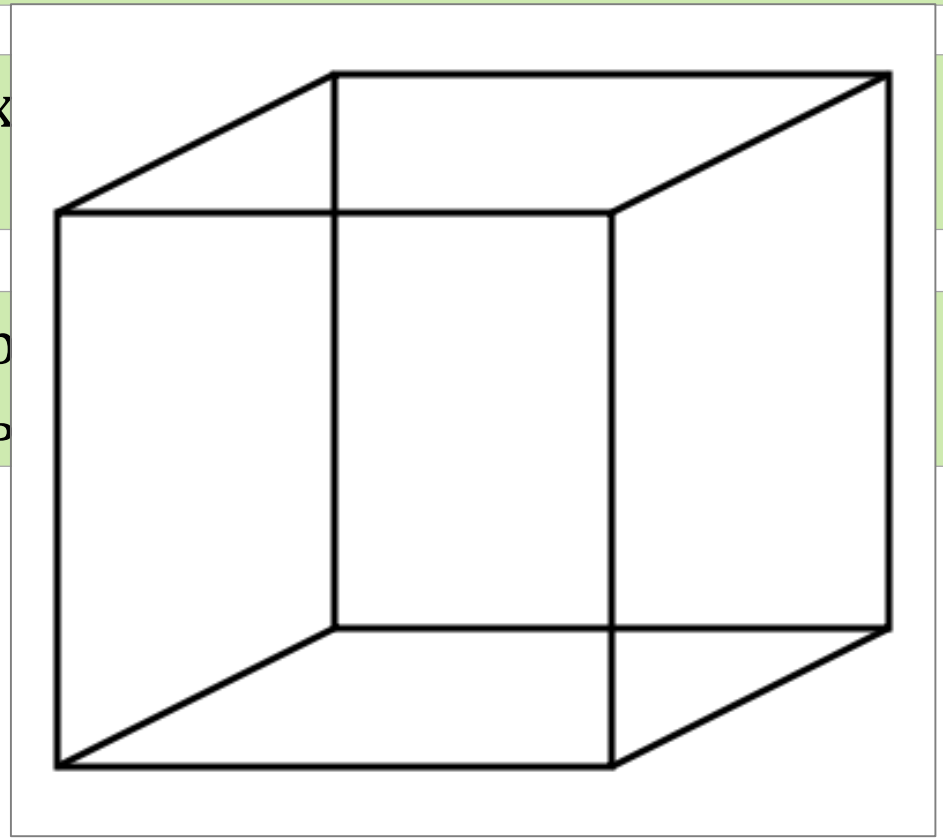
Опр. 5: Избыточные (проверочные) символы – вводимые в исходную последовательность и служащие для обнаружения и исправления ошибок.

ОПРЕДЕЛЕНИЯ

Опр. 6: Непрерывными называются такие коды, в которых введение избыточных символов в кодируемую последовательность информационных символов осуществляется непрерывно, без разделения ее на независимые блоки.

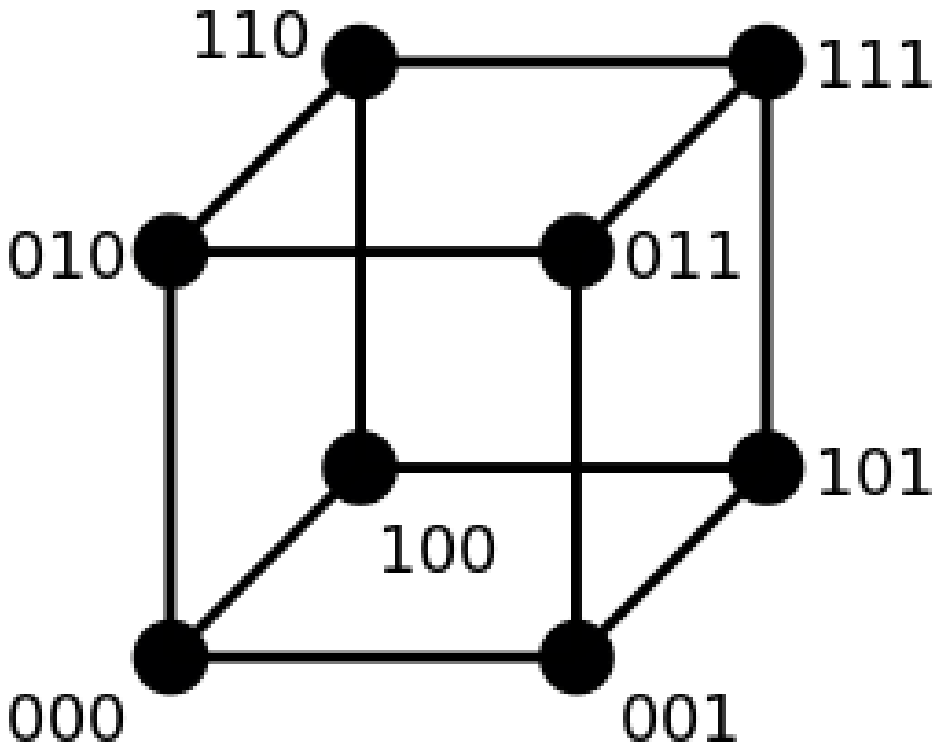
Опр. 7: Разделимые коды – такие, в которых
информационные и проверочные.

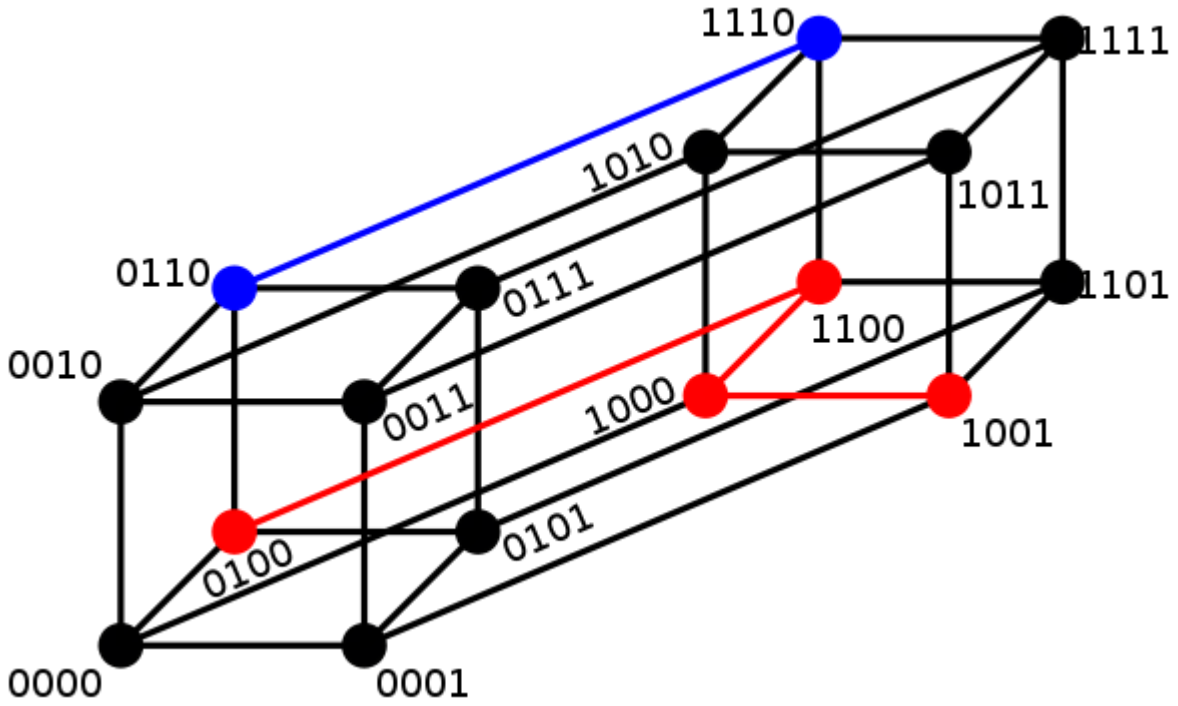
Опр. 8: Неразделимые коды – такие, в котор
разделить на информационные и проверочны



РАССТОЯНИЕ ХЭММИНГА

Опр. 9: Расстоянием Хэмминга (кодовым расстоянием) $d(b_1, b_2)$ между двумя двоичными словами b_1 и b_2 называется количество позиций, в которых эти слова различаются.





РАССТОЯНИЕ ХЭММИНГА

Теорема об обнаруживающем коде. *Пусть при передаче кодовых слов происходит не более k ошибок. Для того чтобы код являлся обнаруживающим, необходимо и достаточно, чтобы наименьшее расстояние между кодовыми словами удовлетворяло неравенству:* $d \geq k + 1$.

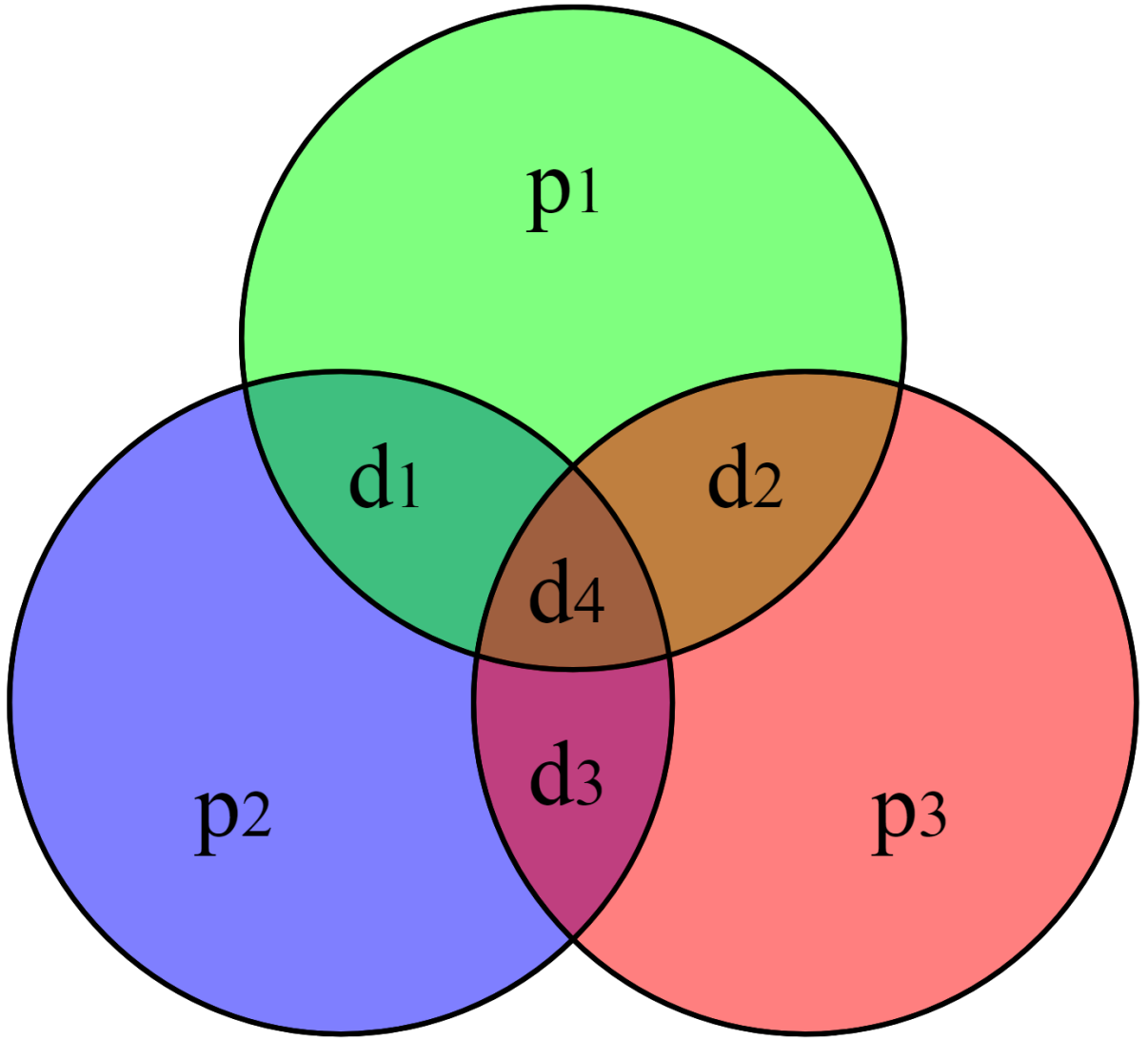
Теорема о корректирующем коде. *Пусть при передаче кодовых слов происходит не более k ошибок. Для того чтобы код являлся корректирующим, необходимо и достаточно, чтобы наименьшее расстояние между кодовыми словами удовлетворяло неравенству:* $d \geq 2k + 1$.

ОБОЗНАЧЕНИЕ КОДОВ

Опр. 10: Если для слов длины m в алфавите B={0, 1} используются кодовые слова длины n, то такие коды называются **(n, m)**–кодами.

Опр. 11: Линейным (матричным) кодом называется код, который можно задать в виде матрицы **G** порядка **m x n**, где кодовое слово β для исходного слова **α** получается как: $\beta = \alpha G$. Матрица **G** называется **порождающей** матрицей.

КОД ХЭММИНГА



КОД ХЭММИНГА

Алгоритм 1: Построение кода Хэмминга

1. Пусть дано m – количество информационных символов. Из неравенства $2^r \geq n$ определяем r – количество проверочных символов.
2. Конструируем код: проверочные символы вставляем на места, номера которых являются степенями двойки: 1, 2, 4, 8, 16 и т.д. На оставшиеся позиции копируем информационные символы в исходном порядке.
3. Под получившимся словом рисуем таблицу размером m на n, где i-тая строка соответствует i-тому проверочному символу.
4. Рассмотрим i-тую строку: в ячейке ставим 1, если соответствующий проверочный бит за нее отвечает. Правило: проверочный бит с номером N контролирует все последующие N бит через каждые N бит, начиная с позиции N
5. Умножение i-той строки на исходное сообщение дает значение i-го проверочного бита.
6. Повторяем шаги 4 и 5 для всех проверочных бит и получаем код.

КОД ХЭММИНГА

Алгоритм 2: Обнаружение ошибки в коде Хэмминга

1. По заданному сообщению длины n определяем r и m.
2. Повторяем алгоритм 3, только значения проверочных битов выписываем отдельно, а не в кодовое слово.
3. Пусть в результате получилась последовательность $b_1 b_2 \dots b_r$ (ее называют **синдромом**)
   1. Если в синдроме все биты равны 0, то в слове нет ошибок.
   2. Иначе есть ошибка. При этом синдром есть перевернутая двоичная запись номера позиции слова, в котором произошла ошибка.

КОД ХЭММИНГА

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
r_0	r_1	x_1	r_2	x_2	x_3	x_4	r_3	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	r_4	x_{12}	x_{13}	x_{14}	x_{15}	
0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	0	1	
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	r_0
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	r_1
0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	r_2
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	r_3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	r_4

КОД ХЭММИНГА

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coverage	p1	✗		✗		✗		✗		✗		✗		✗		✗		✗		✗		…
	p2		✗	✗			✗	✗			✗	✗			✗	✗			✗	✗		
	p4				✗	✗	✗	✗					✗	✗	✗	✗					✗	
	p8								✗	✗	✗	✗	✗	✗	✗	✗						
	p16																✗	✗	✗	✗	✗	

ПОЛИНОМИАЛЬНЫЕ КОДЫ

Опр. 12: Двоичный полином определяется следующим образом. Пусть $a = a_0 \dots a_{m-1}$ – двоичное сообщение. Тогда сопоставим ему многочлен a(x) = $a_0 + a_1 x + a_2 x^2 + \cdots + a_{m-1} x^{m-1}$. Все вычисления происходят по модулю 2.

Опр. 13: CRC (Cyclical Redundancy Check) – циклический избыточный код, вычисляемый посредством деления многочлена, соответствующего исходному сообщению (полином-сообщение), на фиксированный многочлен (полином-генератор). Остаток от такого деления и есть код **CRC**.
CRC-n код использует для построения остатка полином-генератор **n**-й степени.

УПРАЖНЕНИЯ

1. Подсчитать расстояние Хэмминга для слов 001011 и 011001, 01011010 и 00001000
2. Подсчитать минимальное кодовое расстояние для следующего кода:

   01101000 00101111 01010101 11111111
3. Сколько ошибок может обнаружить код из упражнения 2, и сколько может исправить?
4. Пусть дана матрица $G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$. Напишите коды для сообщений 001, 111, 101, 010
5. Напишите систему линейных уравнений для кодов из упражнения 4
6. Выпишите кодирующую матрицу для (7,4)-кода Хэмминга.
7. Постройте код Хэмминга для слова 1010001010.
8. Определите, где произошла ошибка в коде Хэмминга 11110110001011110001
9. Используя полином-генератор $x^4 + 1$, построить CRC-4 код для сообщений 10000000 и 101111001.
10. Проверить, были ли ошибки в коде 100101101001, используя полином-генератор из п.9

УПРАЖНЕНИЯ

11. Для кодирующих матриц $\mathrm{E}_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$ $E_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
    1. построить, соответственно (5,2)-код и (4,3)-код
    2. найти основные характеристики полученных кодов: минимальное расстояние между словами кода; максимальную кратность ошибок, до которой, включительно, они все исправляются или обнаруживаются