

Матричное кодирование Циклические избыточные КОДЫ

Параграфы 22, 27

Матричное кодирование

Пусть E матрица размерности $m \times n$,

состоящая из элементов e_{ij} ,

где i — это номер строки, а j — номер

столбца. Каждый из элементов матрицы e_{ij} может быть либо **0**, либо **1**.

Кодирование реализуется операцией

$$b = aE \text{ или } b_j = a_1e_{1j} + a_2e_{2j} + \dots + a_me_{mj},$$

где кодовые слова рассматриваются как векторы, т.е как матрицы-строки размера $1 \times n$.

Пример.

Рассмотрим следующую 3×6 -матрицу:

$$E = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Тогда кодирование задается такими отображениями: $000 \rightarrow 000000$, $001 \rightarrow 001111$, $010 \rightarrow 010011$, $011 \rightarrow 011100$, $100 \rightarrow 100110$, $101 \rightarrow 101001$, $110 \rightarrow 110101$, $111 \rightarrow 111010$.

Если потребуется закодировать сообщение вида 011011010010 , то сначала делим его на блоки по три символа

Дана исходная информация: $011 \ 011 \ 010 \ 010$

1. Далее записать все возможные комбинации
2. Кодировем каждый блок отдельно

$011100 \ 011100 \ 010011 \ 010011$

Ответ: закодированная информация будет иметь вид –
 $011100 \ 011100 \ 010011 \ 010011$

Кодирование не должно приписывать одно и то же кодовое слово разным исходным сообщениям.

Простой способ добиться этого состоит в том, чтобы m столбцов (в предыдущем примере — первых) матрицы E образовывали единичную матрицу.

При умножении любого вектора на единичную матрицу получается этот же самый вектор, следовательно, разным векторам-сообщениям будут соответствовать разные вектора систематического кода.

Циклические избыточные коды

- Циклический избыточный код (Cyclical Redundancy Check—CRC) имеет фиксированную длину и используется для обнаружения ошибок.
- Наибольшее распространения получили коды CRC-16 и CRC-32, имеющие длину 16 и 32 бита соответственно. Код CRC строится по исходному сообщению произвольной длины, т.е. этот код не является блочным в строгом смысле этого слова. Но при каждом конкретном применении этот код — блочный, $(m, m + 16)$ -код для CRC-16 или $(m, m + 32)$ -код для CRC-32.

- Вычисление значения кода CRC происходит посредством деления многочлена, соответствующего исходному сообщению (полином сообщение), на фиксированный многочлен (полином-генератор). Остаток от такого деления и есть код CRC, соответствующий исходному сообщению. Для кода CRC-16 полином-генератор имеет степень 16, а для CRC-32 — 32. Полиномы-генераторы подбираются специальным образом и для кодов CRC-16/32 стандартизированы Международным консультативным комитетом по телеграфной и телефонной связи (ССИТТ). Для CRC-16, например, стандартным является полином-генератор $x^{16} + x^{12} + x^5 + 1$.

• Пример построения **CRC-4** кода для

сообщения **11010111**,

используя **ПОЛИНОМ-генератор**

$x^4+x^3+x^2+1$.

Исходному сообщению соответствует

полином **$x^7+x^6+x^4+x^2+x+1$,**

т.е. **нумерация битов здесь начинается
справа.**

решение

$$\begin{array}{r|l} x^7 + x^6 + x^4 + x^2 + x + 1 & x^4 + x^3 + x^2 + 1 \\ x^7 + x^6 + x^5 + x^3 & \hline x^5 + x^4 + x^3 + x^2 + x + 1 & \\ x^5 + x^4 + x^3 + x & \\ \hline x^2 + 1 & \end{array}$$

Полиному $x^2 + 1$ соответствуют биты 0101 — это и есть CRC-4

Существуют быстрые алгоритмы для расчета CRC-кодов, использующие специальные таблицы, а не деление многочленов с остатком.

CRC-коды способны обнаруживать одиночную ошибку в любой позиции и, кроме того, многочисленные комбинации кратных ошибок, расположенных близко друг от друга. При реальной передаче или хранении информации ошибки обычно группируются на некотором участке, а не распределяются равномерно по всей длине данных.

Применение

Таким образом, хотя для идеального случая двоичного симметричного канала CRC-коды не имеют никаких теоретических преимуществ по сравнению, например, с простыми контрольными суммами, для реальных систем эти коды являются очень полезными.

Коды CRC используются очень широко: модемами, телекоммуникационными программами, программами архивации и проверки целостности данных и многими другими программными и аппаратными компонентами вычислительных систем.

Домашнее задание

- Законспектировать лекцию.
- Выполнить матричное кодирование своего исходного сообщения (размер исходного сообщения должен соответствовать образцу).
- Подготовить конспекты всех лекций для проверки (на каждой странице подписать ручкой свою фамилию и инициалы, сделать **выделения тем и определений**, сфотографировать и **отправить в личный кабинет через сайт esstu не позднее 30.10**).