Циклические коды. Код CRC.

Выполнила: студентка группы 5104
Ондар Ю.
Проверила: Гармаева.О.А

г. Улан-Удэ
2015 г

Содержание

** Блочные и линейные коды

** CRC код

-- CRC код. Базовые определения

-- Вычисление CRC

-- Коррекция ошибок

Блочные и линейные коды

* Блоковый код представляет собой множество последовательностей символов, именуемых кодовыми словами.

* В общем случае элементы кодового слова выбираются из алфавита с q элементами. На практике наиболее широко используются коды, содержащие два элемента (О и 1), такие коды называются двоичными (q = 2).

*** Относительно большой класс блоковых кодов составляют линейные коды. Для них по сравнению с общим случаем блоковых кодов значительно упрощается операция декодирования.**

*** Линейные коды представляют собой подпространство *Vk* линейного пространства *VJ* и обладают следующим важным свойством: сумма (определенная для этого пространства) двух кодовых слов также является кодовым словом.**

CRC код. Базовые определения

** Cyclic Redundancy Code (CRC) – циклический избыточный код **

* CRC – это значение, которое вычисляется для некоторого блока данных, например, для каждого файла во время архивации.

* При развертывании файла архиватор сравнивает это значение со вновь вычисленным CRC распакованного файла.

* Если они совпадают, то существует очень большая вероятность того, что этот новый файл получился идентичным исходному. При использовании CRC32 вероятность пропустить изменение данных составляет всего 2^{-32}.

* Основная идея состоит в том, чтобы представить файл, как одну огромную строку бит, и поделить ее на некоторое число; Оставшийся в результате остаток и есть CRC.

* Всегда будет оставаться остаток (правда, иногда он может оказаться равным нулю), который лишь на один бит меньше делителя

Пример: 9/3=3, остаток = 0; (9+2)/3=3, остаток = 2.

Вычисление CRC

*** Вычисление CRC использует особый вид вычитания и сложения, своего рода "новую арифметику". Компьютер "забывает" делать перенос при вычислении каждого бита.**

*** Заём при вычислении также "забывают" сделать. Это напоминает операцию "Исключающее ИЛИ" (eXclusive OR, или более привычно - XOR), чем оно фактически и является.**

*** Пример CRC-арифметики:**

```
1001/1111000\1110      9/120\14 остаток 6 (120/9=14 и остаток 6)
     1001    -
     ----
      1100
      1001    -
      ----
       1010
       1001    -
       ----
        0110
        0000    -
        ----
         110 -> остаток
```

* Для вычисления CRC нам необходимо выбрать делитель, который с этого момента мы будет называть полиномом. Степень полинома (W – Width) – это номер позиции его старшего бита, следовательно, полином 1001 будет иметь степень "3", а не "4".

* Обратите внимание, что старший бит всегда должен быть равен 1, следовательно, после того, как Вы выбрали степень полинома, Вам необходимо подобрать лишь значения младших его битов.

РЕЗУЛЬТАТ: 1101011011111

Здесь необходимо упомянуть 2 важных момента:

1. Операция XOR выполняется только в том случае, когда старший бит последовательности равен “1”, в противном случае мы просто сдвигаем ее на один бит влево.

2. Операция XOR выполняется только с младшими W битами, так как старший бит всегда в результате дает “0”.

Коррекция ошибок

*** Процедура коррекции ошибок предполагает два совмещенные процесса: обнаружение ошибки и определение места. После решения этих двух задач, исправление тривиально – надо инвертировать значение ошибочного бита.**

*** В наземных каналах связи, где вероятность ошибки невелика, обычно используется метод детектирования ошибок и повторной пересылки фрагмента, содержащего дефект. Для спутниковых каналов с большими задержками системы коррекции ошибок становятся привлекательными.**

* Код Хэмминга представляет собой блочный код, который позволяет выявить и исправить ошибочно переданный бит в пределах переданного блока. Обычно код Хэмминга характеризуется двумя целыми числами, например, (11,7) используемый при передаче 7-битных ASCII-кодов.

* Такая запись говорит, что при передаче 7-битного кода используется 4 контрольных бита (7+4=11). При этом предполагается, что имела место ошибка в одном бите и что ошибка в двух или более битах существенно менее вероятна.

* При этом предполагается, что имела место ошибка в одном бите и что ошибка в двух или более битах существенно менее вероятна. С учетом этого исправление ошибки осуществляется с определенной вероятностью. Например, пусть возможны следующие правильные коды (все они, кроме первого и последнего, отстоят друг от друга на расстояние 4):

```
00000000
11110000
00001111
11111111
```

* При получении кода 00000111 не трудно предположить, что правильное значение полученного кода равно 00001111. Другие коды отстоят от полученного на большее расстояние Хэмминга.

* Рассмотрим пример передачи кода буквы s = 0x073 = 1110011 с использованием кода Хэмминга (11,7).

Позиция бита:	11	10	9	8	7	6	5	4	3	2	1
Значение бита:	1	1	1	*	0	0	1	*	1	*	*

* Символами “*” помечены четыре позиции, где должны размещаться контрольные биты. Эти позиции определяются целой степенью 2 (1, 2, 4, 8 и т.д.).

* Контрольная сумма формируется путем выполнения операции XOR (исключающее ИЛИ) над кодами позиций ненулевых битов. В данном случае это 11, 10, 9, 5 и 3. Вычислим контрольную сумму:

```
11 = 1011
10 = 1010
09 = 1001
05 = 0101
03 = 0011
 s = 1110
```

* Таким образом, приемник получит код:

Позиция бита:	11	10	9	8	7	6	5	4	3	2	1
Значение бита:	1	1	1	1	0	0	1	1	1	1	0

*** Просуммируем снова коды позиций ненулевых битов и получим нуль:**

11 =	1011
10 =	1010
09 =	1001
08 =	1000
05 =	0101
04 =	0100
03 =	0011
02 =	0010
s =	0000

*** Ну а теперь рассмотрим два случая ошибок в одном из битов посылки, например, в бите 7 (1 вместо 0) и в бите 5 (0 вместо 1). Просуммируем коды позиций ненулевых бит еще раз.**

11 =	1011	11 =	1011
10 =	1010	10 =	1010
09 =	1001	09 =	1001
08 =	1000	08 =	1000
07 =	0111	04 =	0100
05 =	0101	03 =	0011
04 =	0100	02 =	0010
03 =	0011	σ =	0001
02 =	0010		
σ =	0111		

* В общем случае код имеет N=M+C бит и предполагается, что не более чем один бит в коде может иметь ошибку. Тогда возможно N+1 состояние кода (правильное состояние и n ошибочных). Пусть M=4, а N=7, тогда слово-сообщение будет иметь вид: M4, M3, M2, C3, M1, C2, C1. Теперь попытаемся вычислить значения C1, C2, C3. Для этого используются уравнения, где все операции представляют собой сложение по модулю 2:

$$C1 = M1 + M2 + M4$$
$$C2 = M1 + M3 + M4$$
$$C3 = M2 + M3 + M4$$

* Для определения того, доставлено ли сообщение без ошибок, вычисляем следующие выражения (сложение по модулю 2):

$$C11 = C1 + M4 + M2 + M1$$
$$C12 = C2 + M4 + M3 + M1$$
$$C13 = C3 + M4 + M3 + M2$$

*** Результат вычисления интерпретируется следующим образом:**

C11	C12	C13	Значение
1	2	4	Позиция бит
0	0	0	Ошибок нет
0	0	1	Бит С3 не верен
0	1	0	Бит С2 не верен
0	1	1	Бит М3 не верен
1	0	0	Бит С1 не верен
1	0	1	Бит М2 не верен
1	1	0	Бит М1 не верен
1	1	1	Бит М4 не верен

*** Описанная схема легко переносится на любое число n и M. Число возможных кодовых комбинаций M помехоустойчивого кода делится на n классов, где N – число разрешенных кодов. Разделение на классы осуществляется так, чтобы в каждый класс вошел один разрешенный код и ближайшие к нем запрещенные коды.**

*** Если код принят с ошибкой, он заменяется ближайшим разрешенным кодом. При этом предполагается, что кратность ошибки не более q_m.**

* Можно доказать, что для исправления ошибок с кратностью не более q_m m (как правило, оно выбирается равным D = $2q_m$ + 1). В теории кодирования существуют следующие оценки максимального числа N n-разрядных кодов с расстоянием D.

d=1	$n=2^n$
d=2	$n=2^{n-1}$
d=3	$n\,2^n/(1+n)$
d=2q+1	$N \le 2^n (1+\sum_{i=1}^{d} C_n^i)^{-1}$

* Для кода Хэмминга это неравенство превращается в равенство. В случае кода Хэмминга первые k разрядов используются в качестве информационных, причем k=n - $\log_2(n+1)$, откуда следует (логарифм по основанию 2), что k может принимать значения 0, 1, 4, 11, 26, 57 и т.д., это и определяет соответствующие коды Хэмминга (3,1); (7,4); (15,11); (31,26); (63,57) и т.д..

Спасибо за внимание!