Декодирование кода Голея

ДЗ от Иванова — часть вторая

Полиномы, многочлены и биты

В теории кодирования принято оперировать многочленами (полиномами). В домашнем задании используется двоичный код Голея, поэтому коэффициенты полинома 0 или 1. Например, $1x^5+1x^4+0x^3+1x^2+0x+1=x^5+x^4+x^2+1$. Полином характеризуется своими коэффициентами, поэтому их принято записывать в виде двоичного числа. Приведенному выше полиному соответствует число `110101` или `101011`. В первом случае старшие разряды слева, а втором — справа. Старшие разряды соответствуют коэффициентам полинома при старших степенях. Иванов на лекциях использует обе записи, поэтому иногда возникает путаница. Я далее буду использовать первый вид запись — старшие разряды слева, так привычнее.

Таким образом, представления в виде полинома и двоичного числа эквивалентны. Программисту, наверно, легче оперировать битами, а для построения алгебраической теории нужны полиномы.

Двоичный код Голея

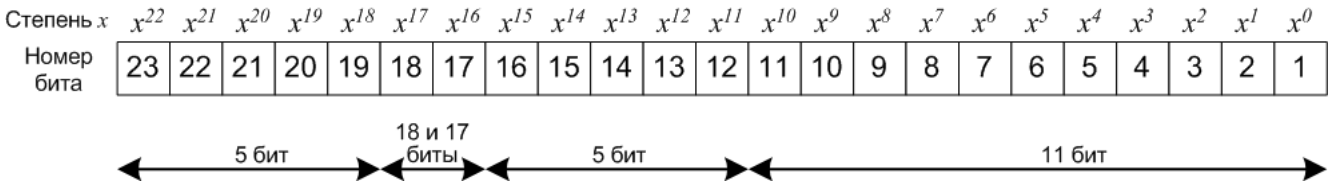
Информационное слово — это двоичное число (полином), которое является полезной информацией. В двоичном коде Голея информационное слово 12 бит. Код Голея определяет, как этим 12 битам сопоставить *кодовое слово* в 23 бита. Такое преобразование вносит избыточность в информацию, но зато позволяет восстановить исходное информационное слово, даже если в кодовое слово внести 3 ошибки. Предположим у нас есть 12 бит полезной информации, мы их кодируем кодом Голея и передаем в канал 23 бита. В канале возникли ошибки и несколько (0, 1, 2, 3) бит передались не правильно. В итоге на приемном конце имеем 23 бита с ошибками. Но поскольку нам известно, что они закодированы кодом Голея, то мы можем точно восстановить 12 бит полезной информации, даже если 3 любых бита из 23 ошибочные. На рисунке представлена иллюстрация сказанного.

Код Голея характеризуется своим *образующим полиномом* — $g(x)=x^{11}+x^{10}+x^6+x^5+x^4+x^2+1$ (`110001110101`). Для получения кодового слова необходимо информационное слово умножить на образующий полином, т.е. перемножить соответствующие полиномы. Получиться полином с максимальной степенью 22, т.е. слово в 23 бита. При умножении полиномов сложение происходи по модулю 2. Для получения информационного слова надо кодовое слово разделить на образующий полином g(x), причем остаток от деления будет равен нулю (делим один полином на другой).

Если в кодовое слово внести ошибки, то остаток от деления на g(x) уже не будет равен нулю. Задача декодирования заключается в поиске ошибочных бит, при исправлении которых остаток от деления на g(x) будет равен нулю.

Декодирование двоичного кода Голея

Прямое декодирование по таблице соответствий очень ресурсоёмкий процесс, поэтому существует алгоритм декодирования «методом ловли ошибок». Введем еще некоторые понятия. *Синдром* — тот самый остаток от деления кодового слова на g(x), кстати, синдром 11 бит. *Вес полинома* — количество единичных бит, обычно говорят про вес синдрома. Теперь сам алгоритм. На рисунке представлена структура кодового слова, используемая в алгоритме.



1. Делим кодовое слово на g(x), т.е. находим синдром. Если вес синдрома равен **нулю** (т.е. g(x) делит кодовое слово без остатка) — это означает, что в кодовом слове нет ошибок. Если вес синдрома **1**, **2** или **3** — это значит, что все (1, 2 или 3) ошибки в младших 11 битах кодового слова. Причем, единичные биты в синдроме соответствуют ошибочным битам в кодовом слове. Если вес больше 3, переходим к шагу 2.

2. Предполагаем, что в 18 бите ошибка, остальные (0, 1 или 2) ошибки в младших 11 битах. Для проверки предположения находи *модифицированный синдром*. Для этого складываем «нормальный» синдром с остатком от деления x^{17} на g(x). Остаток заранее посчитан: `11011001100` (старшие биты слева). Операция сложения эквивалентна побитовой операции **xor** (исключающее или). Если вес модифицированного синдрома равен **0**, **1** или **2** — это значит, что в 18 бите ошибка, а все остальные ошибки в младших 11 битах. Причем, единичные биты в модифицированном синдроме соответствуют ошибочным битам в кодовом слове. Если вес больше 3, переходим к шагу 3.

3. Предполагаем, что в 17 бите ошибка, остальные (0, 1 или 2) ошибки в младших 11 битах. Далее аналогично 2 шагу. Остаток от деления `01101100110` (старшие биты слева). Если вес больше 3, переходим к шагу 4.

4. Делаем циклический сдвиг кодового слова в сторону младших разрядов и начинаем с 1 шага. Когда на очередном этапе ошибки будут найдены, необходимо кодовое слово циклически сдвинуть в обратную сторону, на соответствующее количество бит.

Чтобы каждый раз на первом шаге не делить кодовое слово на g(x), существует правило для нахождения нового синдрома, после циклического сдвига кодового слова. Если в младшем разряде синдрома 0, то новый синдром получается простым циклическим сдвигом старого синдрома. Если в младшем разряде синдрома 1, то для нахождения нового синдрома нужно старый синдром сложить с g(x), а потом результат сдвинуть на 1 бит (не циклически) в сторону младших разрядов.

Далее приведен пример решения на 4-х страницах.

Сахаров К.В. 09.01.2008, Версия 0.4

Пример декодирования с семинара 23.10.2007

```
Декодирование кода Голея
c 100100111001111110001101 кодовое слово с ошибкой
g 110001110101 образующий полином

Решение
Найдем синдром
Делим 100100111001111110001101 на 110001110101

   100100111001111110001101    частное (столбец снизу)
  +110001110101
  =0101010011001               000000000001
   +110001110101
   =0110111011001              000000000011
    +110001110101
    =0001101011001             000000000111
      0011010110011            000000001110
       0110101100110           000000011100
       +110001110101
       =0001000100110          000000111001
         0010001001100         000001110010
          0100010011001        000011100100
          +110001110101
          =0100111011001       000111001001
           +110001110101
           =0101101011000      001110010011
            +110001110101
            =0111001011011     011100100111
             +110001110101
             =001000101110     111001001111
s  01000101110 синдром - остаток от деления

============================================================
шаг  0
   01000101110 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 00101001000 синдром 16

   01000101110 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 10011100010 синдром 17

итого:
синдром     вес
001000101110 5  синдром
000101001000 3  синдром 16
010011100010 5  синдром 17
Ошибки не найдены, переходим к следующему шагу.

============================================================
шаг  1
 100100111001111110001101 старое кодовое слово
 110010011100111111000110 новое  кодовое слово, полученное циклическим сдвигом

 01000101110 старый синдром
 00100010111 новый синдром, получен циклическим сдвигом старого

   00100010111 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 01001110001 синдром 16

   00100010111 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 11111011011 синдром 17
```

```
итого:
синдром     вес
000100010111 5  синдром
001001110001 5  синдром 16
011111011011 9  синдром 17
Ошибки не найдены, переходим к следующему шагу.

============================================================
шаг  2
 1100100111001111000110 старое кодовое слово
 0110010011100111100011 новое  кодовое слово, полученное циклическим сдвигом

 00100010111 старый синдром
В синдроме в младшем разряде 1, поэтому новый вычисляем следующем образом
    00100010111 старый синдром
 + 110001110101 образующий полином g(x)
 = 110101100010
сдвигаем
 =  11010110001 новый синдром

   11010110001 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 10111010111 синдром 16

   11010110001 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 00001111101 синдром 17

итого:
синдром     вес
011010110001 6  синдром
010111010111 8  синдром 16
000001111101 6  синдром 17
Ошибки не найдены, переходим к следующему шагу.

============================================================
шаг  3
 0110010011100111100011 старое кодовое слово
 1011001001110011110001 новое  кодовое слово, полученное циклическим сдвигом

 11010110001 старый синдром
В синдроме в младшем разряде 1, поэтому новый вычисляем следующем образом
    11010110001 старый синдром
 + 110001110101 образующий полином g(x)
 = 101011000100
сдвигаем
 =  10101100010 новый синдром

   10101100010 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 11000000100 синдром 16

   10101100010 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 01110101110 синдром 17

итого:
синдром     вес
010101100010 5  синдром
011000000100 3  синдром 16
001110101110 7  синдром 17
Ошибки не найдены, переходим к следующему шагу.
```

```
============================================================
шаг  4
 10110010011100111110001 старое кодовое слово
 11011001001110011111000 новое  кодовое слово, полученное циклическим сдвигом

 10101100010 старый синдром
 01010110001 новый синдром, получен циклическим сдвигом старого

   01010110001 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 00111010111 синдром 16

   01010110001 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 10001111101 синдром 17

итого:
синдром     вес
001010110001 5  синдром
000111010111 7  синдром 16
010001111101 7  синдром 17
Ошибки не найдены, переходим к следующему шагу.

============================================================
шаг  5
 11011001001110011111000 старое кодовое слово
 01101100100111001111100 новое  кодовое слово, полученное циклическим сдвигом

 01010110001 старый синдром
В синдроме в младшем разряде 1, поэтому новый вычисляем следующем образом
    01010110001 старый синдром
 + 110001110101 образующий полином g(x)
 = 111011000100
сдвигаем
 =  11101100010 новый синдром

   11101100010 синдром
 + 01101100110 остаток от деления x^16 на g(x)
 = 10000000100 синдром 16

   11101100010 синдром
 + 11011001100 остаток от деления x^17 на g(x)
 = 00110101110 синдром 17

итого:
синдром     вес
011101100010 6  синдром
010000000100 2  синдром 16
000110101110 6  синдром 17

============================================================
Вес синдрома 16 меньше либо равен 2, следовательно, в 17 разряде (x^16) ошибка,
а остальные ошибки в младших 11 разрядах.
Исправляем ошибки
   01101100100111001111100 кодовое слово
 +             10000000100 синдром 16
 + 00000010000000000000000 ошибка в 17 разряде
 = 01101110100101001111000 кодовое слово без ошибок

Для получения исходного кодового слова выполняем обратный циклический сдвиг.
  11010010100111100001101 результат декодирования - кодовое слово без ошибок
  01000001000000010000000 внесенная ошибка
  10010011100111110001101 кодовое слово с ошибкой (дано)
```

Разделим результат декодирования на образующий полином.
Делим 11010010100111100001101 на 110001110101

```
   11010010100111100001101    частное (столбец снизу)
  +110001110101
  =0001010111001              000000000001
    0010101110011             000000000010
     0101011100111            000000000100
     +110001110101
     =0110100100100           000000001001
      +110001110101
      =0001010100010          000000010011
        0010101000100         000000100110
         0101010001000        000001001100
         +110001110101
         =0110111111011       000010011001
          +110001110101
          =0001100011101      000100110011
            0011000111010     001001100110
             0110001110101    010011001100
             +110001110101
             =000000000000    100110011001
```

Частное - информационное слово.
100110011001 информационное слово