

Оглавление

Оглавление	1
1 Код Голея	2
1.1 Кодирование	2
1.2 Декодирование	3
2 Задача	5
3 Приложения	6
3.1 Пример работы алгоритма	6
3.2 Входные вектора для кодирования	7
3.3 Входные вектора для декодирования	7
3.4 Умножение матриц	8

Глава 1

Код Голея

Двоичный код Голея - один из связанных друг с другом исправляющих ошибки линейных кодов:

- совершенный двоичный код Голея с параметрами $[27, 12, 7]$;
- расширенный двоичный код Голея, получаемый из совершенного добавлением бита четности, параметры: $[24, 12, 8]$

Мы будем использовать совершенный код Голея, так как в нашем случае он оптимальнее.

В этом случае 12 входных бит данных кодируются в 23 выходных бита.

Данный код позволяет исправлять любые три ошибки в слове.

1.1. Кодирование

Кодирование примитивное: умножение вектора на матрицу.

$$D_{enc} = D_{in} \times M_{cre}, \quad (1.1)$$

где $D_{in}[1 : 12]$ - входной вектор данных, $M_{cre}[12 : 23]$ - порождающая матрица, $D_{enc}[1 : 23]$ - выходной вектор.

Входной вектор D_{in} имеет вид $[D_1 \ D_2 \ \dots \ D_{12}]$, где D_i - исходные двоичные данные.

Порождающая матрица M_{cre} выглядит следующим образом:

$$M_{cre} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

После кодирования получает вектор $D_{enc} = [D_1 D_2 \dots D_{12} \quad \alpha_1 \alpha_2 \dots \alpha_{11}]$, где D_i - исходные биты данных, α_j - проверочные биты данных.

Вектор D_{enc} передаем по зашумленному каналу.

1.2. Декодирование

Декодирование Голя также сводится к умножению матриц с некоторыми усложнениями (формулы 1.2, 1.3).

$$S = D_{in} \times M_{ch} \quad (1.2)$$

$$D_{dec} = D_{in} \oplus Err[S], \quad (1.3)$$

где $D_{in}[1 : 23]$ - вектор входных данных (возможно с ошибками), $D_{dec}[1 : 12]$ - вектор декодированных данных, Err - таблица корректирующих кодов, $M_{ch}[23 : 11]$ - проверочная матрица, $S[1 : 11]$ - вектор синдромов.

Произведение входного вектора данных на проверочную матрицу дает вектор синдромов размерностью $[1:11]$ (формула 1.2). Вектор синдромов является бинарным представлением индекса корректирующего кода в таблице Err. Таблица корректирующих кодов Err содержит 2048 записей длиной 12 бит каждая. Таблица ошибок Err известна и находится в приложенном файле **errors.dat**.

Проверочная матрица M_{ch} выглядит следующим образом:

$$M_{ch} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Для завершения декодирования необходимо входной вектор сложить по модулю 2 с корректирующим (формула 1.3). Если ошибок менее трех, то исходный вектор D_{in} будет равен D_{dec} . При количестве ошибок 4 и более декодирование также будет произведено, но результат будет непредсказуем, так как совершенный код Голя позволяет исправлять до 3х ошибок, но не позволяет обнаруживать 4 и более ошибок.

Глава 2

Задача

Задача сводится к написанию кодера и декодера совершенного кода Голя по алгоритму, описанному в главе 1. Работа описаного алгоритма с промежуточными данными находится в разделе 3.1.

Проверка работы кодера будет осуществляться по известным входным векторам из раздела 3.2

Проверка работы декодера будет осуществляться по известным векторам из раздела 3.3

Глава 3

Приложения

3.1. Пример работы алгоритма

```
Input:      0x00000571 (0b00000000000000010101110001)
Encoded:    0x000AB571 (0b00001010101011010101110001)
Errors:     0x00100008 (0b000100000000000000000001000)
Received:   0x001AB579 (0b00011010101011010101111001)
Syndrome:   0x0000063F (0b00000000000000011000111111)
Correction: 0x00100008 (0b000100000000000000000001000)
Decoded:    0x00000571 (0b00000000000000010101110001)
Result:     Ok
```

```
Input:      0x00000319 (0b00000000000000001100011001)
Encoded:    0x0074B319 (0b011101001011001100011001)
Errors:     0x0000000C (0b000000000000000000000001100)
Received:   0x0074B315 (0b011101001011001100010101)
Syndrome:   0x000000AE (0b0000000000000000010101110)
Correction: 0x0000000C (0b000000000000000000000001100)
Decoded:    0x00000319 (0b00000000000000001100011001)
Result:     Ok
```

```
Input:      0x00000908 (0b000000000000000100100001000)
Encoded:    0x00303908 (0b0011000000011100100001000)
Errors:     0x000408C8 (0b000001000000100011001000)
Received:   0x003431C0 (0b001101000011000111000000)
Syndrome:   0x000000EF (0b0000000000000000011101111)
Correction: 0x0000C200 (0b00000000011000010000000000)
Decoded:    0x000003C0 (0b00000000000000001111000000)
Result:     Error
```

3.2. Входные вектора для кодирования

Двоичное представление входных векторов для кодирования:

```
0b00000000000000010100011001
0b000000000000000100101110111
0b00000000000000011010110010
0b00000000000000010000111100
0b000000000000000111111101100
0b000000000000000111000100010
0b0000000000000001110110110
0b000000000000000110111011100
0b0000000000000001111111010
0b000000000000000110010111010
```

Пример оформления результатов:

```
Input:      0x00000908 (0b0000000000000100100001000)
Encoded:    0x00303908 (0b0011000000011100100001000)

Input:      0x00000319 (0b0000000000000001100011001)
Encoded:    0x0074B319 (0b011101001011001100011001)
```

3.3. Входные вектора для декодирования

Двоичное представление векторов для декодирования:

```
0b000110011001001000100100
0b011110111111100011100001
0b001001110101100101010011
0b011011101000011010110011
0b000111000001001110010111
0b010100000010100010010000
0b000101011000101001000010
0b001001100000010000101100
0b011100101011011001001111
0b011110001001000111111000
```

Пример оформления результатов:

```
Received:    0x0074B315 (0b011101001011001100010101)
Syndrome:    0x000000AE (0b0000000000000000010101110)
Correction:  0x0000000C (0b0000000000000000000001100)
```

```
Decoded:      0x00000319 (0b0000000000000001100011001)

Received:     0x003431C0 (0b001101000011000111000000)
Syndrome:     0x000000EF (0b0000000000000000011101111)
Correction:   0x0000C200 (0b000000001100001000000000)
Decoded:      0x000003C0 (0b0000000000000001111000000)
```

3.4. Умножение матриц

Все представленные числа в матрицах, векторах являются двоичными. Таким образом вся используемая арифметика также является двоичной.

Двоичное умножение матриц отличается от арифметического тем, что вместо арифметической суммы берется сумма по модулю 2.