

Криптография и корректирующее кодирование

12 января 2017 г.

Оглавление

| | | |
|----------|--|-----------|
| 1 | Корректирующие коды, код Хемминга | 3 |
| 1.1 | Общие определения | 3 |
| 1.2 | Расстояние Хемминга и исправление ошибок | 3 |
| 1.3 | Граница Хемминга | 4 |
| 1.4 | Граница Варшамова-Гильберта | 6 |
| 1.5 | Граница Плоткина | 6 |
| 1.6 | Асимптотика границ | 8 |
| 2 | Матрицы и коды Адамара | 10 |
| 2.1 | Матрицы и коды Адамара, общее представление | 10 |
| 2.2 | Построение матрицы Адамара по способу Пэли | 11 |
| 3 | Линейные коды | 12 |
| 3.1 | Базовые факты, коды Адамара | 12 |
| 3.2 | Смежные классы и декодирование по синдрому | 13 |
| 3.3 | Полиномиальные коды | 14 |
| 3.4 | Совершенные линейные коды | 14 |
| 3.5 | Двоичные циклические коды | 15 |
| 3.5.1 | Свойства циклического кода | 15 |
| 3.5.2 | Порождающая и проверочная матрицы циклического кода | 16 |
| 3.6 | Модификации линейных кодов | 17 |
| 3.7 | Бинарные коды Голея | 18 |
| 3.8 | Бинарные CRC-коды | 19 |
| 4 | Регистры сдвига и линейная сложность | 21 |
| 4.1 | Регистры сдвига с линейной обратной связью | 21 |
| 4.2 | Линейная сложность, алгоритм Берлекэмп-Мэсси | 22 |
| 4.3 | Порождение симплексного кода с помощью регистра сдвига | 22 |
| 5 | Булевы функции и их свойства | 23 |
| 5.1 | Определения. Алгебраическая нормальная форма | 23 |
| 5.1.1 | Алгебраическая нормальная форма | 23 |
| 5.1.2 | Быстрое преобразование Мёбиуса | 24 |
| 5.2 | Коды Рида-Маллера | 25 |
| 5.2.1 | Взаимосвязь кодов Рида-Маллера разных порядков | 25 |
| 5.2.2 | Выколотые коды Рида-Маллера | 26 |
| 5.3 | Декодирование кода $\mathcal{R}(1, m)$ | 26 |

Глава 1

Корректирующие коды, код Хемминга

1.1 Общие определения

Кодируется последовательность бит. При **непрерывном коде** кодируется вся последовательность, при **блочном** последовательность разбивается на блоки по k бит и каждый блок кодируется отдельно.

Определение 1.1. Инъективное отображение $f : K \rightarrow \{0, 1\}^n$, $K \subset \{0, 1\}^k$ называется кодом. Образ любого слова из $\{0, 1\}^k$ называется кодовым словом или кодом. Множество $C = f(\{0, 1\}^k)$ также называется кодом.

Определение 1.2. Код называется раздельным, если $[n] = A \cup B$, $A \cap B = \emptyset$, $|A| = k$ и $\forall x \in K : f(x)|_A = x$, то есть, для некоторого подмножества бит кода оно совпадает с прообразом как строка. Биты множества A называются информационными, а из множества B — проверочными.

Определение 1.3. Код называется линейным, если соответствующее отображение f линейно.

Определение 1.4. Раздельный код называется систематическим, если проверочные символы являются линейной комбинацией информационных. То же самое, что раздельный линейный код.

Определение 1.5. Два кода f и g назовем эквивалентными, если $g(x) = f(\pi(x))$, где $\pi(x)$ — это x под действием некоторой перестановки π .

Определение 1.6. Скорость кода $C \subset \{0, 1\}^n$ — это величина $R = \frac{1}{n} \log_2 |C|$. При $|C| = 2^k$ имеет место $R = \frac{k}{n}$.
Избыточность кода — это величина $1 - R$

1.2 Расстояние Хемминга и исправление ошибок

Определение 1.7. Расстоянием Хемминга между строками $x, y \in \{0, 1\}^n$ будем называть величину

$$d(x, y) = |\{i : x_i \neq y_i\}|$$

Определение 1.8. $d(C) = \min_{\substack{x, y \in C \\ x \neq y}} d(x, y)$ — кодовое расстояние кода C .

Обозначение: (n, k, d) -код, код с длиной кодируемого слова k , кодового слова n и минимальным кодовым расстоянием d . $[n, K, d]$ -код — код с длиной кодового слова n , количеством слов K и минимальным кодовым расстоянием d .

Определение 1.9. Код обнаруживает ошибки в r битах, если существует отображение $g : \{0, 1\}^n \rightarrow \{0, 1\}$, такое, что $\forall x \in \{0, 1\}^k, |z| \leq r : g(f(x) \oplus z) = 1$

Определение 1.10. Код исправляет ошибки в r битах, если существует отображение $g : \{0, 1\}^n \rightarrow \{0, 1\}^k$, такое, что $\forall x \in \{0, 1\}^k, |z| \leq r : g(f(x) \oplus z) = x$

Теорема 1.1. Для того, чтобы код C позволял обнаружить ошибки в r битах, необходимо и достаточно, чтобы $d(C) \geq r + 1$

Теорема 1.2. Для того, чтобы код C позволял исправить ошибки в r битах, необходимо и достаточно, чтобы $d(C) \geq 2r + 1$

Доказательство. \Leftarrow

$g(x) = \arg \min_{y \in \{0,1\}^k} d(x, f(y))$. Пусть $x = f(y) + z$ и $|z| \leq r$ и $g(x) \neq y$. Тогда $d(f(g(x)), x) \leq r$, а, значит $d(f(y), f(g(x))) \leq d(x, f(y)) + d(x, f(g(x))) \leq 2r$. Противоречие.

\Rightarrow

Рассмотрим $x, y \in C$ такие, что $d(x, y) \leq 2r$. Тогда легко видеть, что существует z , такое, что $d(x, z) \leq r$ и $d(y, z) \leq r$. Тогда, как бы мы не определили $g(z)$, мы получим противоречие с x или y . \square

1.3 Граница Хемминга

Определение 1.11. Шаром радиуса r с центром в x назовем множество точек

$$B_r(x) = \{y : d(x, y) \leq r\}$$

Количество вершин в шаре в пространстве $\{0, 1\}^n$ обозначим $S_r(n)$

Замечание 1.1. $S_r(x) = \sum_{i=0}^r C_n^i$.

Доказательство. $S_r(n) = |B_r(0)|$. Строки в $B_r(0)$ — это строки с не более чем r единичными битами. \square

Определение 1.12. Энтропией дискретной случайной величины ξ принимающей значения $1, \dots, n$ с вероятностями p_1, \dots, p_n называется

$$H(\xi) = - \sum_{i=1}^n p_i \log_2(p_i)$$

Лемма 1.1.

$$\frac{1}{n+1} 2^{nH(\frac{r}{n})} \leq C_n^r \leq 2^{nH(\frac{r}{n})}$$

Доказательство. По формуле Стирлинга

$$C_n^r \simeq \frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} \cdot \frac{n^n}{k^k (n-k)^{n-k}}$$

С другой стороны

$$2^{nH(\frac{r}{n})} = 2^{n \left(-\frac{r}{n} \log_2 \frac{r}{n} - (1-\frac{r}{n}) \log_2 (1-\frac{r}{n}) \right)} = \frac{\left(\frac{r}{n} \right)^{-r}}{\left(1 - \frac{r}{n} \right)^{n-r}} = \frac{n^n}{r^r (n-r)^{n-r}}$$

Тогда для достаточно больших n достаточно показать

$$\frac{1}{n+1} \leq \frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} \leq 1$$

Второе неравенство очевидно, поскольку в знаменателе квадратичная зависимость.

$$\frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} = \sqrt{\frac{n}{k(n-k)}} \frac{1}{\sqrt{2\pi}}$$

$k(n-k) \leq \frac{n^2}{4}$, тогда имеем

$$\frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} \geq \frac{2}{\sqrt{2\pi n}}$$

для достаточно больших n последнее $\geq \frac{1}{n+1}$

\square

Теорема 1.3. Для достаточно больших n и при условии $0 < r \leq \frac{n}{2}$ верно

$$\frac{\log_2 S_r(n)}{n} = H\left(\frac{r}{n}\right) + O\left(\frac{\log_2 n}{n}\right)$$

где $H\left(\frac{r}{n}\right)$ — энтропия случайной величины, принимающей значения 0 и 1 с вероятностями $\frac{r}{n}$ и $1 - \frac{r}{n}$.

Доказательство. Покажем, что при $r \leq \frac{n}{2}$ наибольшим слагаемым будет C_n^r .

$$\frac{C_n^i}{C_n^{i+1}} = \frac{n!(i+1)!(n-i-1)!}{n!i!(n-i)!} = \frac{i+1}{n-i}$$

Возрастание C_n^i равносильно $\frac{C_n^i}{C_n^{i+1}} \leq 1 \iff i+1 \leq n-i \iff 2i \leq n-1$. То есть C_n^i больше предыдущего сочетания, если $2(i-1) \leq n-1$ то есть $i \leq \frac{n+1}{2}$. Тогда имеем

$$C_n^r \leq S_r(n) \leq (r+1)C_n^r$$

Воспользуемся леммой, прологарифмируем формулу оттуда:

$$-\log_2(n+1) + nH\left(\frac{r}{n}\right) \leq \log_2 S_r(n) \leq \log_2(r+1) + nH\left(\frac{r}{n}\right)$$

Поделим три части на n

$$-\frac{\log_2(n+1)}{n} + H\left(\frac{r}{n}\right) \leq \frac{\log_2 S_r(n)}{n} \leq \frac{\log_2(r+1)}{n} + H\left(\frac{r}{n}\right)$$

Тогда получили, что требовалось,

$$\frac{\log_2 S_r(n)}{n} = H\left(\frac{r}{n}\right) + \underbrace{c}_{|c| \leq 1} \frac{\log_2(r+1)}{n}$$

□

Теорема 1.4. (Граница Хемминга) Для любого (n, k) -кода, исправляющего r ошибок верно

$$n - k \geq \log_2 \left(\sum_{i=0}^r C_n^i \right)$$

Доказательство. Рассмотрим прообразы исправляющей функции g . $g^{-1}(y)$. По определению $|g^{-1}(y)| \geq S_r(n)$ и $y_1 \neq y_2 \implies g^{-1}(y_1) \cap g^{-1}(y_2) = \emptyset$. Тогда для завершения доказательства достаточно расписать

$$2^n = |\{0, 1\}^n| = \left| \bigcup_{y \in \{0, 1\}^k} g^{-1}(y) \right| \geq \sum_{y \in \{0, 1\}^k} S_r(k) = 2^k S_r(n)$$

□

Теорема 1.5. Если $n - k \geq \log_2(n+1)$, то существует $(n, k, 3)$ код, то есть, граница Хемминга достигается.

Доказательство. Построим явно такой линейный код. $C = \{Hx = 0\}$, где H — матрица $(n-k) \times n$. Пусть H_{ij} — это i -й бит числа j ($1 \leq i \leq n-k$; $1 \leq j \leq n$). Заметим, что в условиях теоремы в матрице нет двух одинаковых столбцов, то есть, ее ранг не меньше 2. Пусть существуют $x, y \in C$, такие, что $d(x, y) \leq 2$ тогда $d(0, x \oplus y) \leq 2$. То есть $x \oplus y$ имеет не более двух единиц в двоичной записи $H(x \oplus y) = H_{j_1} \oplus H_{j_2} = 0$, что противоречит выводу о ранге. Тогда кодовое расстояние полученного кода равно 3. □

Пример 1.1. Построим систематический (n, k) код Хемминга.

Пусть $a \in \{0, 1\}^k$; $b \in \{0, 1\}^n$. Кодирование преобразование $E(a) = b$. Наложим следующие ограничения:

$$\begin{cases} b_i = a_i & i \leq k \\ b_{i+k} = (\Gamma_i, a) & i \leq n-k \end{cases}$$

То есть $b = a(E_k | \Gamma^T)$. То есть, мы построили порождающую матрицу кодирующей функции. Построим теперь проверочную матрицу:

$$b_{i+k} = (\Gamma_i, (b_1, \dots, b_k)) \iff b_{i+k} \oplus (\Gamma_i, (b_1, \dots, b_k)) = 0$$

То есть, $H = (\Gamma|E_{n-k})$. Условие $Hb = 0$ является необходимым и достаточным для того, чтобы b являлось кодом, поскольку образом такого b является (b_1, \dots, b_k) .

Если столбцы матрицы H различны, то по 1.5 мы можем исправлять одну ошибку. Давайте построим явно исправляющую функцию.

Пусть $b' = b \oplus e_i$, где $e_i = (\underbrace{0, \dots, 0}_{i-1}, 1, 0, \dots, 0)$. Тогда $Hb' = H_i - i$ — столбец матрицы H . Так как все столбцы различны, мы можем узнать, в каком бите была ошибка. Hb' называется *синдромом* вектора b' .

1.4 Граница Варшамова-Гильберта

Теорема 1.6. *Существует (n, k) -код с минимальным расстоянием d , такой, что*

$$n - k \leq \log_2 S_{d-1}(n)$$

Доказательство. Выберем точку c_1 . Рассмотрим $B_{d-1}(c_1)$ и пометим точки в нем. Пока есть непомяченные точки будем выбирать c_i и помечать точки в шаре $B_{d-1}(C_i)$. Так мы построим последовательность точек c_1, \dots, c_K , такую, что $i \neq j \implies d(c_i, c_j) \geq d$. Все точки $\{0, 1\}^n$ покрыты хотя бы одним шаром, то есть $K \cdot S_{d-1}(n) \geq 2^n$. $K \geq 2$, если $d - 1 < n$, так как $d((0, \dots, 0), (1, \dots, 1)) = n$. Выберем $k = \lceil \log_2 K \rceil$, тогда $2^k S_{d-1}(n) \geq 2^n \implies S_{d-1}(n) \geq 2^{n-k}$. \square

Следствие 1.1. *Существует (n, k) -код, исправляющий r ошибок и удовлетворяющий*

$$n - k \leq \log_2(S_{2r}(n))$$

Замечание 1.2. Мы получили верхнюю границу на количество исправляющих символов. Граница Хемминга — нижняя граница, то есть

$$\log_2 S_r(n) \leq n - k \leq \log_2 S_{2r}(n)$$

1.5 Граница Плоткина

Теорема 1.7. *Для $[n, K, d]$ -кода выполнено $d \leq \frac{n \cdot \frac{K}{2}}{K-1}$. В частности, для (n, k, d) -кода верно $d \leq \frac{n 2^{k-1}}{2^k - 1}$*

Доказательство. Рассмотрим $D = \sum_{x, y \in C} d(x, y)$. С одной стороны

$$D \geq 2_K^2 d = K(K-1)d$$

С другой стороны, рассмотрим каждый бит строк и обозначим

$$d_i(x, y) = \begin{cases} 0 & x_i = y_i \\ 1 & x_i \neq y_i \end{cases}$$

Тогда $d(x, y) = d_1(x, y) + \dots + d_n(x, y)$. Тогда

$$D = \sum_{i=1}^n \underbrace{\sum_{x, y \in C} d_i(x, y)}_{D_i}$$

Заметим, что

$$D_i = 2|\{x \in C : x_i = 0\}| \cdot |\{x \in C : x_i = 1\}|$$

Тогда $D_i \leq 2\left(\frac{K}{2}\right)^2$, а, значит

$$D \leq \frac{nK^2}{2}$$

Таким образом,

$$\frac{nK^2}{2} \geq K(K-1)d \iff \frac{nK}{K-1} \geq d$$

\square

Теорема 1.8. Если существует, (n, k) -код C , такой, что $d(C) \geq \frac{n}{2}$, то

$$k \leq \log_2(2n) \iff \frac{\overbrace{K}^{2^k}}{2} \leq n$$

Доказательство. Рассмотрим преобразование

$$\underbrace{(b_1, \dots, b_n)}_{\in \{0,1\}^n} \mapsto ((-1)^{b_1}, \dots, (-1)^{b_n})$$

Пусть $v^{(1)}, \dots, v^{(K)}$ — векторы, полученные этим преобразованием из векторов кода. $d(b^{(i)}, b^{(j)}) \geq \frac{n}{2} \iff (v^{(i)}, v^{(j)}) \leq 0$.

Пусть $\frac{K}{2} > n$, тогда покажем, что не может существовать набора $v^{(1)}, \dots, v^{(K)}$ с требуемым свойством. Рассмотрим $x \in \mathbb{R}^n$, такой, что $(x, v^{(i)}) \neq 0$ для всех i . Например, можно рассмотреть $(1, 0, \dots, 0)$.

Тогда $(x, v^{(i)}) > 0$ для не менее чем $\frac{K}{2}$ векторов, либо $(x, v^{(i)}) < 0$ для не менее чем $\frac{K}{2}$ векторов. НУО верно первое иначе рассмотрим $-x$.

Тогда у нас есть набор из $\frac{K}{2} > n$ векторов, таких, что $(x, v^{(i)}) > 0$ для всех i . Количество векторов превышает n , тогда

$$\exists \lambda: \sum_{i=1}^{n+1} \lambda_i v^{(i)} = 0$$

НУО $\exists \lambda_i > 0$, иначе поменяем знак всем λ , тогда обозначим $I = \{i: \lambda_i > 0\} \neq \emptyset$. Можем записать

$$\sum_{i=1}^{n+1} \lambda_i v^{(i)} = \underbrace{\sum_{i \in I} \lambda_i v^{(i)}}_z + \sum_{i \notin I} \lambda_i v^{(i)} = 0$$

- $z \neq 0$. Тогда $(z, z) > 0$, с другой стороны

$$(z, 0 - z) = \left(\sum_{i \in I} \lambda_i v^{(i)}, - \sum_{i \notin I} \lambda_i v^{(i)} \right) = - \sum_{\substack{i \in I \\ j \notin I}} \underbrace{\lambda_i}_{>0} \underbrace{\lambda_j}_{<0} \overbrace{(v^{(i)}, v^{(j)})}^{<0} \leq 0$$

Получаем противоречие

- $z = 0$. Тогда $(z, x) = 0$, но

$$(z, x) = \left(\sum_{i \in I} \lambda_i v^{(i)}, x \right) = \sum_{i \in I} \underbrace{\lambda_i}_{>0} \underbrace{(v^{(i)}, x)}_{>0} > 0$$

□

Теорема 1.9. Для (n, k) кода, такого, что $n \geq 2d(C)$ выполнено

$$n - k \geq 2d(C) - \log_2 4d(C)$$

Доказательство. При $n = 2d$ воспользуемся 1.8 и получим $-k \geq -\log_2(2n)$ и прибавим к обеим частям $n = 2d$

При $n > 2d$ обозначим $n = 2d + t$ и рассмотрим два случая:

1. $t \geq k$. Тогда сразу $n \geq 2d + k$ и теорема доказана
2. $t < k$. Тогда выберем в коде t информационных символов I_0 тогда рассмотрим код $C' = \{x|_{[n] \setminus I_0} : x \in C \wedge x|_{I_0} = a\}$ для произвольного $a \in \{0, 1\}^t$. Кодовое расстояние этого кода не менее d , поскольку мы вычеркивали одинаковые символы, $n' = 2d$. Тогда $k - t \leq \log_2(2n')$. Тогда

$$n - k = 2d - (k - t) \geq 2d - \log_2(4d)$$

□

1.6 Асимптотика границ

$R = \frac{k}{n}$ — скорость кода.

Обозначим $\delta(C) = \frac{d(C)}{n}$ — относительное кодовое расстояние.

Обозначим $\mathcal{U} = \{(R, \delta)\} \subset [0, 1] \times [0, 1]$ множество пар, таких, что существует последовательность (n_i, k_i, d_i) кодов, таких, что

$$\begin{aligned} n_i &\rightarrow \infty \\ \frac{k_i}{n_i} &\rightarrow R \\ \frac{d_i}{n_i} &\rightarrow \delta \end{aligned}$$

Оценим величину $\bar{R}(\delta) = \sup\{R: (R, \delta) \in \mathcal{U}\}$

Замечание 1.3. При $\delta > \frac{1}{2}$ $\bar{R}(\delta) = 0$

Доказательство.

$$d \leq \frac{n2^{k-1}}{2^k - 1} \implies \delta + \frac{O(1)}{n} \leq \frac{2^{k-1}}{2^k - 1}$$

При $n \rightarrow \infty$ получим (пользуясь $2\delta - 1 > 0$) $2^k \leq \frac{2\delta}{2\delta - 1}$, тогда $k \leq \log_2 \frac{2\delta}{2\delta - 1}$, и значит $R = \frac{k}{n} \rightarrow 0$ □

Утверждение 1.1. $\bar{R}(\delta) \leq 1 - H(\frac{\delta}{2})$

Доказательство. $n - k \geq \log_2 S_{\lfloor \frac{d(C)-1}{2} \rfloor}(n)$ известно из теоремы о границе Хемминга. $d(C) = \lfloor \delta n \rfloor$ имеем

$$1 - \frac{k}{n} \geq \frac{\log_2 S_{\lfloor \frac{\lfloor n\delta \rfloor - 1}{2} \rfloor}(n)}{n}$$

По следствию

$$\frac{\log_2 S_r(n)}{n} = H\left(\frac{r}{n}\right) + O\left(\frac{\log_2 n}{n}\right)$$

тогда

$$1 - R \geq O\left(\frac{\log_2 n}{n}\right) + H\left(\frac{\lfloor n\delta \rfloor - 1}{2}\right)$$

пренебрегая округлениями

$$R + O\left(\frac{\log_2 n}{n}\right) \leq 1 - H\left(\frac{\delta}{2}\right)$$

и при $n \rightarrow \infty$

$$\bar{R}(\delta) \leq 1 - H\left(\frac{\delta}{2}\right)$$

□

Утверждение 1.2. $\bar{R}(\delta) \geq 1 - H(\delta)$ при $\delta \leq \frac{1}{2}$

Доказательство. Из теоремы о границе Варшавова-Гильберта знаем, что

$$n - k \leq \log_2 S_{d-1}(n)$$

в нашем случае

$$1 - \frac{k}{n} \leq \frac{\log_2 S_{\lfloor n\delta \rfloor - 1}(n)}{n}$$

по следствию из теоремы о границе Хемминга

$$1 - \frac{k}{n} \leq H\left(\frac{\lfloor n\delta \rfloor - 1}{n}\right) - O\left(\frac{\log_2 n}{n}\right)$$

тогда при $n \rightarrow \infty$ получаем требуемое. □

Утверждение 1.3. $\bar{R}(\delta) \leq 1 - 2\delta$ при $\delta \leq \frac{1}{2}$

Доказательство. Из последней теоремы о границе Плоткина

$$n - k \geq 2n\delta - \log_2(4n\delta)$$

можно переписать как

$$\frac{k}{n} \leq 1 - 2\delta + \frac{\log_2 4n\delta}{n}$$

тогда при $n \rightarrow \infty$ имеем $\bar{R} \leq 1 - 2\delta$

□

Глава 2

Матрицы и коды Адамара

2.1 Матрицы и коды Адамара, общее представление

Определение 2.1. Матрицей Адамара называется матрица $H \in \{-1, 1\}^{n \times n}$, такая, что $H \cdot H^T = nE_n$.

Матрица адамана в нормализованном виде — это матрица, у которой первая строка и первый столбец состоят из единиц.

Двоичная матрица Адамара, это матрица, полученная из матрицы Адамара заменой -1 на 1 а 1 на 0 .

Утверждение 2.1. Умножение строчки или столбца матрицы Адамара на -1 переводит ее в матрицу Адамара.

Доказательство. Умножение строчки или столбца на единицу, это доножение слева или справа на матрицу $d = \text{diag}(1, \dots, 1, -1, 1, \dots, 1)$. Тогда в первом случае

$$(dH) \cdot (dH)^T = dHH^T d^T = d(nE)d^T = nEdd^T = nE$$

а во втором

$$(Hd) \cdot (Hd)^T = Hdd^T H^T = HH^T = nE$$

□

Теорема 2.1. Если существует матрица Адамара порядка n , то $n \in \{1, 2\} \cup \{4k\}$

Доказательство. Пусть $n \geq 3$ и существует H . Тогда представим ее в нормализованном виде и разделим столбцы на четыре типа:

1. Начинается с $(1, 1, 1)$ — i штук
2. Начинается с $(1, 1, -1)$ — j штук
3. Начинается с $(1, -1, 1)$ — k штук
4. Начинается с $(1, -1, -1)$ — l штук

Запишем условия ортогональности строк $(1, 2)$, $(2, 3)$ и $(1, 3)$:

$$\begin{cases} i + j - k - l = 0 \\ i - j + k - l = 0 \\ i - j - k + l = 0 \end{cases}$$

Тогда $i = j = k = l$, тогда $n = 4i$

□

Утверждение 2.2. Если H — матрица Адамара, то

$$\begin{pmatrix} H & H \\ H & -H \end{pmatrix}$$

— тоже матрица Адамара.

Доказательство.

$$\begin{pmatrix} H & H \\ H & -H \end{pmatrix} \cdot \begin{pmatrix} H^T & H^T \\ H^T & -H^T \end{pmatrix} = \begin{pmatrix} HH^T + HH^T & HH^T - HH^T \\ HH^T - HH^T & HH^T + HH^T \end{pmatrix} = 2nE_{2n}$$

□

Такие матрицы Адамара называются матрицами Сильвестра.

Определение 2.2. Симплексным кодом Адамара называется $[K-1, K, \frac{K}{2}]$ -код, состоящий из строк двоичной матрицы Адамара из которой удален первый столбец.

Утверждение 2.3. Для симплексного кода Адамара выполнено $K = \frac{2d}{2d-n}$.

Доказательство. Очевидно.

□

Замечание 2.1. Если матрица Адамара, построена по способу Сильвестра, то симплексный код, построенный по ней, линейен.

2.2 Построение матрицы Адамара по способу Пэли

Определение 2.3. Пусть $p \in \mathbb{P} \setminus \{2\}$. $\{a \in \{0, \dots, p-1\} : \exists b: b^2 = a\}$ называется множеством квадратичных вычетов.

Определение 2.4. Функция

$$\chi(i) = \begin{cases} 0 & i \text{ кратно } p \\ 1 & i \pmod p \text{ вычет} \\ -1 & i \pmod p \text{ невычет} \end{cases}$$

называется символом Лежандра.

Теорема 2.2. $\forall c \neq 0 \pmod p$ выполнено $\sum_{b=0}^{p-1} \chi(b)\chi(b+c) = -1$

Конструкция 2.1. Матрица Джекобстола. $Q = \{q_{ij}\}_{p \times p}$. $q_{ij} = \chi(j-i)$.

Лемма 2.1. $Q \cdot Q^T = pE - \mathbf{1}_{p \times p}$

$$Q \mathbf{1}_{p \times p} = \mathbf{1}_{p \times p} Q = 0$$

Доказательство. $Q \mathbf{1}_{p \times p} = \mathbf{1}_{p \times p} Q = 0$, так как по модулю p существует $\frac{p-1}{2}$ вычетов и $\frac{p-1}{2}$ невычетов.

Рассмотрим $P = \{p_{ij}\} = Q \cdot Q^T$. Тогда

$$\begin{aligned} p_{ii} &= \sum_{k=0}^{p-1} q_{ik}^2 = p \\ p_{ij} &= \sum_{k=0}^{p-1} q_{ik} q_{jk} \\ p_{ij} &= \sum_{k=0}^{p-1} \chi(i-k) \chi(j-k) = \sum_{k=0}^{p-1} \chi(i-k) + \chi((i-k) + (j-i)) = -1 \end{aligned}$$

□

Лемма 2.2. Пусть

$$H = \begin{pmatrix} 1 & \mathbf{1}_p \\ \mathbf{1}_p & Q - E \end{pmatrix}$$

Тогда H — матрица Адамара

Доказательство.

$$H \cdot H^T = \begin{pmatrix} 1 & \mathbf{1}_p \\ \mathbf{1}_p & Q - E \end{pmatrix} \cdot \begin{pmatrix} 1 & \mathbf{1}_p \\ \mathbf{1}_p & Q^T - E \end{pmatrix} = \begin{pmatrix} p+1 & \mathbf{0}_p \\ \mathbf{0}_p & \mathbf{1}_{p \times p} + (Q-E)(Q^T-E) \end{pmatrix}$$

Распишем

$$\mathbf{1}_{p \times p} + (Q-E)(Q^T-E) = \mathbf{1}_{p \times p} + QQ^T - Q - Q^T + E\mathbf{1}_{p \times p} + QQ^T - Q - Q^T + E$$

заметим, что $q_{ij} = \chi(i-j) = \chi(-1)\chi(j-i) = -\chi(j-i)$, тогда $Q^T = -Q^T$, тогда

$$\mathbf{1}_{p \times p} + QQ^T - Q - Q^T + E = \mathbf{1}_{p \times p} + QQ^T + E = (p+1)E$$

□

Глава 3

Линейные коды

3.1 Базовые факты, коды Адамара

Определение 3.1. Код называется линейным, если множество кодовых слов C является линейным подпространством $\{0, 1\}^n$.

Определение 3.2. Весом Хэмминга $a \in \{0, 1\}^n$ назовем $w(a) = \{i: a_i = 1\}$

Замечание 3.1. $d(a, b) = w(a \oplus b)$

Лемма 3.1. Пусть C — линейный код. Тогда $d(C) = \min_{\substack{x \in C \\ x \neq 0}} w(x)$

Доказательство. $d(C) = \min_{a \neq b \in C} d(a, b) = \min_{a \neq b \in C} w(a \oplus b) = \min_{\substack{x \in C \\ x \neq 0}} w(x)$ □

Определение 3.3. Пусть C — некоторый линейный код с порождающей матрицей G и проверочной матрицей H . Тогда дуальным к нему называется код C^\perp с порождающей матрицей H и проверочной матрицей G .

Если C являлся (n, k) -кодом, то C^\perp будет $(n, n - k)$ -кодом.

Теорема 3.1. Дуальный код Хэмминга $(2^m - 1, 2^m - 1 - m)$ является кодом Адамара с матрицей Сильвестра.

Доказательство. Будем доказывать по индукции.

База: $m = 2$. Тогда $n = 2^m - 1 = 3$, $k = 2^m - 1 - m = 1$. Тогда проверочная матрица такого кода Хэмминга имеет вид $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ Тогда все векторы дуального кода выглядят как: $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$. Этот код совпадает с соответствующим

кодом Адамара.

Переход: пусть доказано для $n = 2^{m-1} - 1$. Пусть $\bar{H} \in \{0, 1\}^{(m-1) \times 2^{m-1}}$ — проверочная матрица для кода Хэмминга $(2^{m-1} - 1, 2^{m-1} - 1 - (m - 1))$.

Покажем, что матрица

$$H = \begin{pmatrix} 0 \dots 0 & 1 & 1 \dots 1 \\ \bar{H} & \mathbf{0}_{m-1} & \bar{H} \end{pmatrix}$$

является проверочной матрицей кода Хэмминга $(2^m - 1, 2^m - 1 - m)$. Это почти очевидно, достаточно заметить, что столбцы матрицы различны и ее размерность $m \times (2^m - 1)$ (следует из того же свойства для \bar{H} и отсутствия в \bar{H} нулевого столбца).

По индукционному предположению матрица \bar{H} порождает строки матрицы \mathcal{A}' — усеченной бинарной матрицы Адамара размера $2^{m-1} \times 2^{m-1} - 1$. Тогда матрица $(\bar{H} | \mathbf{0}_{m-1} | \bar{H})$ порождает строки матрицы $(\mathcal{A}' | \mathbf{0}_{2^{m-1}} | \mathcal{A}')$.

Добавим в $(\bar{H} | \mathbf{0}_{m-1} | \bar{H})$ первую строку H_1 , чтобы получить матрицу H . Тогда можно сделать вывод, что матрица H порождает все строки матрицы $(\mathcal{A}' | \mathbf{0}_{2^{m-1}} | \mathcal{A}')$ и строки, полученные из них прибавлением H_0 . Тогда в итоге мы получим коды

$$\begin{pmatrix} \mathcal{A}' & \mathbf{0}_{2^{m-1}} & \mathcal{A}' \\ \mathcal{A}' & \mathbf{1}_{2^{m-1}} & \mathbf{1} - \mathcal{A}' \end{pmatrix}$$

Припишем слева столбец из нулей и получим, что новая матрица — это в точности матрица, полученная из $(\mathbf{0}_{2^{m-1}} | \mathcal{A}')$ по правилу Сильвестра. Таким образом, теорема доказана. \square

Следствие 3.1. Код Адамара с матрицей Сильвестра является линейным.

Теорема 3.2. Пусть C — линейный код, H — его проверочная матрица.

1. В проверочной матрице H любые $d-1$ столбцов линейно независимы $\iff d(C) \geq d$
2. Если любые $d-1$ столбцов матрицы H линейно независимы и существуют d линейно зависимых столбцов, то $d(C) = d$

Доказательство. \Rightarrow

По лемме $d(C) = \min_{x \in C} w(x)$. Пусть существует $x \in C$ такое, что $w(x) < d$. $Hx = 0$. Пусть i_1, \dots, i_r — номера ненулевых компонент x ($r < d$). Тогда $H_{i_1} \oplus H_{i_2} \oplus \dots \oplus H_{i_r} = 0$, но это противоречит условию линейной независимости столбцов.

\Leftarrow

Если $H_{i_1} \oplus \dots \oplus H_{i_r} = 0$, то рассмотрим вектор $x = \{x_j\}$, $x_j = \begin{cases} 0 & \exists l: j = i_l \\ 1 & \text{иначе} \end{cases}$, Для такого вектора $Hx = 0$, но $w(x) = r < d$.

Пункт 2 непосредственно следует из пункта 1. \square

3.2 Смежные классы и декодирование по синдрому

Определение 3.4. Смежным классом группы G по подгруппе C называется множество вида

$$\begin{aligned} Cb &= \{xb: x \in C\} && \text{правый} \\ bC &= \{bx: x \in C\} && \text{левый} \end{aligned}$$

Определение 3.5. Синдром вектора x относительно линейного кода C с проверочной матрицей H называется вектор Hx

Теорема 3.3. Пусть $x, y \in \{0, 1\}^n$. Тогда $x, y \in Cz$ для некоторого $z \iff Hx = Hy$

Доказательство. $\Rightarrow x = a + z, y = b + z, a, b \in C$. Тогда

$$Hx = Ha + Hz = Hz = Hb + Hz = Hy$$

$$\Leftarrow Hx = Hy \implies H(x + y) = 0, \text{ тогда } x, y \in Cx. \quad \square$$

Пусть $b \in C, b' = b + e$, где e — вектор ошибок. Тогда $Hb' = He$, то есть, ошибку для b' нужно искать в его смежном классе по C .

Лидер — это слово наименьшего веса в смежном классе. Лидер является наиболее вероятным вектором ошибок.

Утверждение 3.1. Будем полагать вектором ошибок лидера соответствующего смежного класса. Составим матрицу $A = \{A_{ij}\}_{2^{n-k} \times 2^k}$, $A_{i,0}$ — лидер смежного класса i , $A_{0,i} \in C$ и $A_{ij} = A_{i,0} \oplus A_{0,j}$.

1. Исправим все ошибки, являющиеся лидерами
2. Для любого слова A_{ij} слово $A_{0,j}$ является ближайшим к A_{ij} кодовым словом.

Доказательство. 1. Очевидно

2. $A_{ij} = A_{0,j} + A_{i,0}$. $A_{i,0}$ — лидер. $d(A_{ij}, A_{0,j}) = w(A_{i,0})$.

Рассмотрим другое кодовое слово $A_{0,j'}$.

$$d(A_{ij}, A_{0,j'}) = w(A_{ij} \oplus A_{0,j'})$$

$$A_{ij} \oplus A_{0,j'} = A_{i,0} \oplus \underbrace{A_{0,j} \oplus A_{0,j'}}_{\in C}$$

Тогда $A_{ij} \oplus A_{0,j'}$ лежит в смежном классе i , значит $w(A_{ij} \oplus A_{0,j'}) \geq w(A_{i,0})$, что и требовалось. \square

3.3 Полиномиальные коды

Определение 3.6. Установим взаимно однозначное соответствие между многочленами степени $< n$ и двоичными векторами из $\{0, 1\}^n$.

$$\sum_{i=0}^{n-1} g_i x^i \mapsto (x_0, \dots, x_{n-1}) \mapsto \sum_{i=0}^{n-1} g_i x^i$$

. Тогда рассмотрим некоторый многочлен $g(x)$, тогда кодовые многочлены получаются по правилу $b(x) = a(x)g(x)$, где $\deg(a(x)) < k$. Тогда, если $\deg(g(x)) = n - k$, то получается (n, k) код.

Пример 3.1. $(6, 4)$ код, с порождающим многочленом $1 + x + x^2$

$$\begin{array}{rcl} 0001 & \xrightarrow{x^3} & 000111 \\ 0010 & \xrightarrow{x^2} & 001110 \\ 0100 & \xrightarrow{x} & 011100 \\ 1000 & \xrightarrow{1} & 111000 \end{array}$$

3.4 Совершенные линейные коды

Определение 3.7. Линейный (n, k) -код, исправляющий r ошибок называется совершенным, если для него достигается граница Хэмминга:

$$2^{n-k} = S_r(n)$$

Замечание 3.2. Для нелинейных кодов граница Хэмминга имеет вид

$$K = \frac{2^n}{S_r(n)}$$

Пример 3.2. $(2m + 1, 1)$ код. Кодовые слова $\begin{pmatrix} 0 & \dots & 0 \\ 1 & \dots & 1 \end{pmatrix}$. Этот код исправляет m ошибок.

$$S_m(2m + 1) = \sum_{i=0}^m C_{2m+1}^i = \frac{1}{2} \sum_{i=0}^m (C_{2m+1}^i + C_{2m+1}^{2m+1-i}) = 2^m$$

Тогда $2^{2m+1-1} = 2^{2m} = S_m(2m + 1)$, что и требуется по определению.

Пример 3.3. Код Хэмминга с $n = 2^m - 1$, $k = 2^m - 1 - m$, $m \geq 2$. Код исправляет одну ошибку, $S_1(n) = 1 + n = 2^m$. Тогда

$$2^{n-k} = 2^{2^m-1-(2^m-1-m)} = 2^m = S_1(n)$$

Теорема 3.4. Следующие условия равносильны

1. Существует двоичный совершенный код C в $\{0, 1\}^n$, который исправляет одну ошибку
2. $n = 2^m - 1$

Доказательство. $2 \implies 1$ Должно выполняться $K = \frac{2^n}{n+1}$. K может быть целым, только если $n + 1 = 2^m$ для некоторого m .

$1 \implies 2$ Доказали в примере 3.3. □

Пример 3.4. $(23, 12)$ -код Голя, исправляющий 3 ошибки. $S_3(23) = 1 + 23 + C_{23}^2 + C_{23}^3 = 2048 = 2^{11}$. Тогда $2^{23} = S_3(23) \cdot 2^{12}$.

3.5 Двоичные циклические коды

3.5.1 Свойства циклического кода

Определение 3.8. Линейный код C называется циклическим, если $\forall b \in C: b^{(1)} \in C$, где $(b_0, \dots, b_{n-1})^{(1)} = (b_{n-1}, b_0, \dots, b_{n-2})$

Аналогично обозначим $b^{(j)} = (b^{(j-1)})^{(1)}$ — сдвиг на j позиций вправо.

Определение 3.9. Кодовым многочленом, соответствующим $b \in C$ назовем многочлен $\sum_{i=0}^{n-1} b_i x^i$

Теорема 3.5. $b^{(j)}(x) = x^j b(x) \pmod{x^n + 1}$

Доказательство. Распишем $x^j b(x)$:

$$x^j b(x) = \sum_{i=0}^{n-j-1} b_i x^{i+j} + \sum_{i=n-j}^{n-1} b_i x^{i+j} = \sum_{i=0}^{n-j-1} b_i x^{i+j} + x^n \underbrace{\sum_{i=n-j}^{n-1} b_i x^{i+j-n}}_{q(x)}$$

Рассмотрим многочлен $q(x) = b_{n-j} + b_{n-j+1}x + \dots + b_{n-1}x^{j-1}$ и прибавм его дважды к $x^j b(x)$ ($q(x) + q(x) = 0$):

$$x^j b(x) = \underbrace{b_{n-j} + b_{n-j+1}x + \dots + b_{n-1}x^{j-1}}_{q(x)} + b_0 x^j + \dots + b_{n-j-1} x^{n-1} + x^n q(x) + q(x)$$

Тогда по модулю $x^n + 1$ получаем $b^{(j)}(x)$ □

Теорема 3.6. В циклическом коде существует только один ненулевой многочлен минимальной степени.

Доказательство. Пусть есть два таких многочлена $q_1(x) = x^m + \dots$; $q_2(x) = x^m + \dots$. Тогда из линейности кода $q_1(x) + q_2(x) \in C(x)$. Но

$$(q_1 + q_2)(x) = \underbrace{x^m + x^m}_{=0} + \underbrace{\dots}_{deg < m}$$

тогда q_1 и q_2 не минимальны по степени. противоречие. □

Определение 3.10. Кодовый многочлен $g(x)$ минимальной степени среди многочленов $C(x)$ называется порождающим многочленом C .

Теорема 3.7. Свободный член $g(x)$ — порождающего многочлена циклического кода, равен 1.

Доказательство. Пусть $g_0 = 0$, тогда $g_1 + g_2 x + \dots + g_{n-1} x^{n-2} \in C(x)$, но его степень меньше, чем у g . Противоречие. □

Теорема 3.8. Пусть $g(x)$ — порождающий многочлен для циклического кода длины n . Тогда $b(x) \in C(x) \iff b(x)$ кратно $g(x)$.

Доказательство. \Leftarrow Пусть $b(x) = g(x) \cdot a(x)$. $deg(a) \leq n - m - 1$, тогда

$$b(x) = g(x) \sum_{i=0}^{n-m-1} a_i x^i = \sum_{i=0}^{n-m-1} a_i \underbrace{g(x) x^i}_{=g^{(i)}(x)}$$

таким образом, $b(x)$ представлен в виде линейной комбинации циклических сдвигов $g(x)$, то есть $b(x) \in C(x)$

\Rightarrow Пусть $b(x) \in C(x)$. Можно записать $b(x) = g(x) \cdot q(x) + r(x)$. Нужно показать, что $r(x) = 0$

$$r(x) = \underbrace{b(x)}_{\in C(x)} + \underbrace{g(x)q(x)}_{\in C(x)}$$

Тогда $r(x) \in C(x)$. $deg(r(x)) < deg(g(x))$, тогда по теореме 3.6 $r(x) = 0$. □

Теорема 3.9. Пусть код порождается многочленом $g(x)$. Тогда следующие условия равносильны

1. C является циклическим

2. $g(x)$ — делитель $x^n + 1$

Доказательство. $1 \implies 2$ Рассмотрим $b \in C$. По теореме 3.5 имеем $b(x)x^j = b^{(j)}(x) + (x^n + 1)q(x)$. Выберем j так, чтобы $\deg(b(x)x^j) = n$, тогда $q(x) = 1$. Тогда

$$\exists j \in \{0, \dots, n-1\}: x^j b(x) = b^{(j)}(x) + (x^n + 1)$$

Так как C циклический и порождается $g(x)$, то $b^{(j)}(x) = g(x)a_j(x)$. Тогда

$$\underbrace{x^j b(x)}_{\text{кратно } g(x)} = \underbrace{b^{(j)}(x)}_{\text{кратно } g(x)} + (x^n + 1)$$

Тогда и $x^n + 1$ кратно $g(x)$.

$2 \implies 1$ Снова запишем

$$x^j b(x) = b^{(j)}(x) + (x^n + 1)q(x)$$

Тогда

$$b^{(j)}(x) = \underbrace{x^j b(x)}_{\text{кратно } g(x)} - \underbrace{(x^n + 1)}_{\text{кратно } g(x)} q(x)$$

Таким образом, код циклический. □

3.5.2 Порождающая и проверочная матрицы циклического кода

Пусть C — циклический код с порождающим многочленом $g(x) = 1 + g_1x + \dots + g_{r-1}x^{r-1} + x^r$. Тогда все кодовые многочлены имеют вид

$$b(x) = g(x) \underbrace{a(x)}_{\deg=k-1} = a_0g(x) + a_1xg(x) + \dots + a_{k-1}x^{k-1}g(x)$$

То есть, любой кодовый многочлен представляется как линейная комбинация многочленов $x^jg(x)$. Тогда порождающая матрица имеет вид:

$$G = \begin{pmatrix} 1 & g_1 & g_2 & \dots & g_{r-1} & 1 & 0 & \dots & 0 \\ 0 & 1 & g_1 & \dots & g_{r-2} & g_{r-1} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 1 & g_1 & \dots & \dots & g_{r-1} & 1 \end{pmatrix}$$

Теперь построим проверочную матрицу. Рассмотрим $h(x)$, такой, что $x^n + 1 = h(x)g(x)$. Тогда рассмотрим произвольный кодовый многочлен $b(x) = q(x)g(x)$.

$$b(x)h(x) = q(x)g(x)h(x) = q(x)(x^n + 1) = q(x) + x^n a(x)$$

Заметим, что $\deg(a(x)) \leq k-1$, а мономы $x^n a(x)$ имеют степень не менее n тогда коэффициенты $b(x)h(x)$ при $x^k, x^{k+1}, \dots, x^{n-1}$ равны нулю. Давайте выразим эти коэффициенты через коэффициенты b и h :

$$\begin{aligned} \sum_{i=0}^k b_i h_{k-i} &= 0 \\ \sum_{i=0}^k b_{i+1} h_{k-i} &= 0 \\ &\dots \end{aligned}$$

Тогда в матричном виде это выглядит как:

$$H = \begin{pmatrix} h_k & h_{k-1} & h_{k-2} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ 0 & h_k & g_{k-1} & \dots & h_2 & h_1 & h_0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & h_k & g_{k-1} & \dots & \dots & h_1 & h_0 \end{pmatrix}$$

Замечание 3.3. Строки в G и H линейно независимы, поскольку у каждой строки есть компонент, отсутствующий во всех строках с большими номерами. Формально можно доказать по индукции.

Замечание 3.4. Порождающим многочленом дуального кода, порожденного многочленом $g(x)$ с проверочным многочленом $h(x)$ является многочлен $x^k h(x^{-1})$.

Доказательство. Многочлен $x^k h(x^{-1}) = h_k + x h_{k-1} + \dots + x^{k-1} h_1 + x^k h_0$, то есть, это многочлен $h(x)$ с развернутыми коэффициентами. Тогда порождающая матрица для этого многочлена совпадает с проверочной для кода, порожденного C . \square

3.6 Модификации линейных кодов

Определение 3.11. $(n+1, k)$ -код, полученный из (n, k) -кода добавлением одного контрольного бита (иначе говоря, дополнительной переменной), называется *расширенным кодом* (*extended code*).

Вообще говоря, добавлять можем любой бит, но это не всегда имеет смысл.

Утверждение 3.2. Любой (n, k, d) -код с нечётным кодовым расстоянием можно расширить до $(n+1, k, d+1)$ -кода добавлением бита проверки чётности.

Доказательство. Если между двумя словами было расстояние d , то одно из них имеет чётный вес, а другое нечётный, т.к. d нечётно. Тогда очевидно, что добавление бита проверки чётности увеличит расстояние между ними. \square

Определение 3.12. $(n-1, k)$ -код, полученный из (n, k) -кода удалением одного из контрольных битов (удалением переменной), называется *проколотым кодом* (*punctured code*).

Если расширим код, а затем уменьшим его на тот же контрольный бит, на который увеличивали, получим исходный код.

Если удаляемый бит принимает значение 1 в кодовом слове минимального веса, то минимальное кодовое расстояние уменьшается.

Определение 3.13. Код, полученный удалением информационных битов, называется *укороченным кодом* (*shortened code*).

Это значит удаление строки из порождающей матрицы и удаление столбца из проверочной. Т.е. (n, k) -код превращается в $(n-1, k-1)$ -код.

Определение 3.14. Код, полученный добавлением информационного бита, называется *удлинённым кодом* (*lengthened code*).

Это значит, что мы добавили строку в порождающую матрицу и столбец в проверочную. Т.е. (n, k) -код превращается в $(n+1, k+1)$ -код.

Утверждение 3.3. При удлинении и при укорочении минимальное кодовое расстояние не меняется.

Доказательство.

1. При удлинении очевидно.
2. При укорочении происходит следующее: из G вычёркивается строка и соответствующий её столбец *единичной подматрицы*. Соответственно, вычёркивается столбец из проверочной матрицы. Любая линейная комбинация строк G имеет вес как минимум d .

$$a_1 g_1 + \dots + a_n g_n \geq d, \forall \{a_i\}$$

Вычёркивание i -ой строки и соответствующего ей столбца — это линейная комбинация с $a_i = 0$.

\square

Определение 3.15. Код, полученный удалением некоторых кодовых слов, называется *суженным кодом* (*expurgated code*).

Возможно построить суженный код так, чтобы он оставался линейным.

Минимальное кодовое расстояние может увеличиться.

Определение 3.16. Код, полученный добавлением новых кодовых слов, называется *дополненным кодом* (*augmented code*).

Пример 3.5. (7, 4)-код Хэмминга.

Построим расширенный код двумя способами: начиная с проверочной матрицы и начиная с порождающей. Новая переменная — дополнительная проверка чётности для всех битов.

1. Проверочная матрица

$$H = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array} \right]$$

Последняя строка соответствует уравнению $\sum_{i=0}^6 x_i = x_7$, то есть x_7 — бит проверки четности.

Линейными преобразованиями получим

$$H = \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array} \right]$$

Ей соответствует порождающая матрица

$$G = \left[\begin{array}{ccc|c|cccc} 1 & 0 & 1 & |1| & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & |0| & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & |1| & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & |1| & 0 & 0 & 0 & 1 \end{array} \right]$$

соответствующая начальной порождающей, к которой добавили 1 столбец (4-ый).

2. Порождающая матрица

$$G = \left[\begin{array}{ccc|c|cccc} 1 & 0 & 1 & |?| & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & |?| & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & |?| & 0 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & |?| & 0 & 0 & 0 & 1 \end{array} \right]$$

. Добавим такой столбец, что количество единиц в каждой строке чётно. Легко видеть, что это тот же столбец, который мы получили в первом случае и других быть не может.

Почему появляется условие чётности по строкам? Вспомним, $G = (\Gamma^t | E)$, $H = (E | \Gamma)$. От H хотим, чтобы линейными преобразованиями над строками можно было получить строку из всех единиц. Поскольку в H есть единичная подматрица, единственный способ это сделать — просуммировать все строки с коэффициентами 1. Тогда нам необходимо, чтобы все столбцы Γ были веса 1, то есть чтобы все строки Γ^t были веса 1. Следовательно, все строки G должны иметь вес 0.

3.7 Бинарные коды Голея

Чтобы бинарный (n, k, d) -код был совершенным, необходимо выполнение условия плотной упаковки:

$$2^k \sum_{k=0}^{\lfloor \frac{d-1}{2} \rfloor} C_n^k = 2^n$$

Голей нашел два возможных кандидата: (23, 12, 7) и (90, 78, 5).

Теорема 3.10. Не существует бинарного (90, 78, 5)-кода.

Доказательство. Предположим, что существует C — (90, 78, 5)-код. Не умаляя общности можем считать, что $0 \in C$ (иначе выберем любой вектор C и прибавим его ко всем векторам кода).

Пусть $Y = \{y \in \{0, 1\}^{90} : y_0 = y_1 = 1 \wedge w(y) = 3\}$. Очевидно, $|Y| = 88$. Так как C — совершенный код, каждому $y \in Y$ соответствует единственный $x \in C$ причем $d(x, y) = 2$. Тогда $1 \leq w(x) \leq 5$, но $d(x, 0) \geq 5$ и тогда мы можем заключить, что $w(x) = 5$. Тогда $y \subset x$, то есть $y_i = 1 \implies x_i = 1$.

Пусть $X = \{x \in C : x_0 = x_1 = 1 \wedge w(x) = 5\}$. Каждому $y \in Y$ соответствует единственный $\phi(y) = x \in X$, такой, что $d(x, y) = 2$. С другой стороны, рассмотрим $x \in X$. Заменяя две из трех единиц, стоящих на позициях $\{2, \dots, 89\}$ на нули мы получим три различных $y_1, y_2, y_3 \in Y$, при этом $d(x, y_1) = d(x, y_2) = d(x, y_3) = 2$, тогда $\phi(y_1) = \phi(y_2) = \phi(y_3) = x$. Для любых других $y \in Y$ $d(x, y) > 2$. Тогда все элементы Y должны разбиться на тройки по значению $\phi(y)$. Но 88 не делится на 3. Противоречие. \square

Определение 3.17. Расширенным кодом Голя назовем $(24, 12)$ -код, построенный с помощью порождающей матрицы $G = (E_{12}|A)$, где A — фиксированная матрица 12×12 , обладающая следующими свойствами:

- $A = A^T$
- $i \neq j \implies A_i \perp A_j$ (где A_i, A_j — строки матрицы A)

Утверждение 3.4. Так построенный код обладает кодовым расстоянием 8 и исправляет три ошибки.

Замечание 3.5. Удалив один проверочный бит из $(24, 12, 8)$ -кода получим $(23, 12, 7)$ -код — совершенный код Голя.

Утверждение 3.5. Код Голя самодуален, то есть $G^\perp = G$

Доказательство. Заметим, что строки матрицы G ортогональны. Это следует из того, что строки E ортогональны и строки A ортогональны. Тогда $G \subset G^\perp$. Но размерности G и G^\perp равны 12, тогда $G = G^\perp$ \square

3.8 Бинарные CRC-коды

Определение 3.18. CRC-код — это циклический код, используемый для *обнаружения* ошибок. Пусть порождающий многочлен нашего кода — $g(x)$. Тогда будем производить кодирование по правилу

$$c(x) = x^k m(x) + (x^k m(x) \bmod g(x))$$

Будем проверять наличие ошибок в $\bar{c}(x)$ следующим образом:

$$\begin{cases} \text{ошибка} & \text{если } \bar{c}(x) \neq 0 \\ \text{принимает} & \text{если } \bar{c}(x) = 0 \end{cases}$$

Обозначим $e(x) = \bar{c}(x) + c(x)$.

Замечание 3.6. В одну сторону наша проверка корректна, если ошибки нет, то мы точно примем вектор. Но мы можем принять и вектор с ошибкой.

Определение 3.19. Вектор ошибок содержит пакет ошибок длины B , если расстояние между первой и последней ошибкой равно B , то есть, существует i , такое, что

$$e(x) = x^i(1 + e_1x + \dots + x^{B-1})$$

Утверждение 3.6. Верны следующие утверждения

1. Ошибка $e(x) = x^i$ для $i \in \{0, \dots, n-1\}$ будет найдена
2. Если $g(x) = (1+x)\bar{g}(x)$, то $\forall e(x): w(e) \bmod 2 = 1$ будет найдена
3. Если $e(x)$ содержит пакет ошибок длины $n-k$, то такая ошибка будет найдена
4. Если $e(x)$ содержит пакет ошибок длины $n-k+1$, то она не будет найдена только если $e(x) = x^i g(x)$
5. Вероятность, что ошибка с блоком длины $l > n-k+1$ не будет найдена равна 2^{k-n}

Доказательство. 1. $g(x) = 1 + \dots + x^r$, $r > 0$. Рассмотрим произвольный многочлен $a(x) = x^{d_l} + \dots + x^{d_r}$. Тогда $g(x)a(x) = x^{d_l} + \dots + x^{r+d_r}$. Следовательно, так как $d_l < r + d_r$, x^{d_l} не может делиться на $g(x)$

2.

$$c(x) = (1+x)\bar{g}(x)m(x) = (1+x)\left(\sum_{i=0}^{n-2} t_i x^i\right) = t_0 + x(t_0 + t_1) + \dots + t_{n-2}x^{n-1}$$

Посчитаем сумму коэффициентов: $t_0 + (t_0 + t_1) + (t_1 + t_2) + \dots + t_{n-2} = 2(\sum_{i=0}^{n-2} t_i) = 0$. То есть, если ошибка содержит нечетное число единиц, то она не поделится на $g(x)$

3. Пусть $e(x)$ содержит пакет ошибок длины $n - k$. Тогда

$$e(x) = x^i(1 + e_1x + \dots + x^{n-k-1})$$

$i \in \{0, \dots, k\}$. Пусть $e(x) = f(x)g(x)$.

$$\deg(f) = \deg(e) - \deg(g) = (i + n - k - 1) - (n - k) = i - 1$$

Тогда вспомним, что g не кратно x^j ни для каких j . Таким образом f кратно x^i , но $\deg(f) < i$. Значит, $e(x)$ не делится на $g(x)$ и мы обнаружим такую ошибку.

4. Если $e(x) = x^i(1 + e_1x + \dots + x^{n-k})$, тогда ошибка может не распознаться только если $e(x) = x^i g(x)$

5. Пусть $e(x)$ содержит пакет ошибок длины $l > n - k + 1$. Тогда можем записать $e(x) = x^i a(x)g(x)$, опять исходя из факта, что g не кратно x^j для всех $j > 0$. Тогда $\deg(a) = l - (n - k) - 1$ и свободный член a равен 1. Тогда возможных вариантов выбора a существует $2^{l-n+k-2}$.

Будем считать e равномерно распределенным по всем возможным ошибкам с блоком длины l . Тогда вероятность того, что такая ошибка поделится на $g(x)$ равна

$$\frac{\overbrace{(n-l+1)}^{\text{выбор } i} 2^{l-n+k-2}}{2^{l-2}(n-l+1)} = 2^{k-n}$$

□

Глава 4

Регистры сдвига и линейная сложность

4.1 Регистры сдвига с линейной обратной связью

Хотим генерировать поток битов из некоторого начального конечного количества. Рассмотрим следующий алгоритм:

Алгоритм 4.1. Имеем s_0, s_1, \dots, s_{l-1} , где l будем называть длиной регистра сдвига. Пусть $f : \{0, 1\}^l \rightarrow \{0, 1\}$. Тогда будем генерировать дальнейшие биты по рекуррентному соотношению $s_i = f(s_{i-1}, s_{i-2}, \dots, s_{i-l})$.

f будем называть функцией обратной связи.

s_i — выход регистра на шаге i .

Рассмотрим регистр сдвига с линейной функцией обратной связи (РСЛОС)

Определение 4.1. Если $f(x_0, \dots, x_{l-1}) = \sum_{i=0}^{l-1} c_i x_i$, то многочлен, ассоциированный с РСЛОС: $c(x) = 1 + c_0 x + \dots + c_{l-1} x^l$.

Определение 4.2. Периодом регистра называется число $\min\{N \in \mathbb{N} : \forall i \geq N \ S_{N+i} = S_i\}$

Свойства:

1. $s_0 = \dots = s_{l-1} = 0 \implies \forall i: s_i = 0$
2. Период регистра конечен.

Доказательство. Если

$$\begin{cases} s_i = s_j \\ s_{i+1} = s_{j+1} \\ \dots \dots \\ s_{i+l-1} = s_{j+l-1} \end{cases}$$

То $s_{i+l} = s_{j+l}$ по определению. Тогда $\forall k \geq 0: s_{i+k} = s_{j+k}$. Таким образом $(s_i, s_{i+1}, \dots) = (s_j, s_{j+1}, \dots)$. Но тогда существует не более 2^l различных типов таких последовательностей.

□

3. $T \leq 2^l - 1$. Непосредственно следует из доказательства предыдущего пункта.
4. $c_{l-1} = 0 \implies$ период начинается не с начала последовательности (c_T не всегда равно c_0)
5. $c_{l-1} = 1 \implies c_T = c_0$
6. $c(x)$ неприводим над $\mathbb{F}_2 \implies 2^l - 1$ кратно T
7. $c(x)$ примитивный над $\mathbb{F}_2 \implies T = 2^l - 1$

Утверждение 4.1. Пусть известны s_i, \dots, s_{i+2l-1} и известно, что регистр имеет длину l . Тогда можно найти регистр сдвига, порождающий такую последовательность.

Доказательство. Составим систему уравнений, относительно c_i :

$$\begin{cases} s_{i+l} = c_0 s_{i+l-1} + c_1 s_{i+l-2} + \dots + c_{l-1} s_0 \\ s_{i+l+1} = c_0 s_{i+l} + c_1 s_{i+l-1} + \dots + c_{l-1} s_1 \\ \dots = \dots \\ s_{i+2l-1} = c_0 s_{i+2l-2} + c_1 s_{i+2l-3} + \dots + c_{l-1} s_{i+l-1} \end{cases}$$

Система совместна по построению s_i , тогда решение — подходящий регистр сдвига. Если уравнения линейно-независимы, регистр сдвига определяется однозначно. \square

4.2 Линейная сложность, алгоритм Берлекэмпа-Мэсси

Определение 4.3. Регистр сдвига порождает последовательность s , если для начальных значений s_0, \dots, s_{l-1} регистр выдает последовательность s .

Определение 4.4. Линейной сложностью последовательности бит (конечной или бесконечной) s назовем

- 0, если $s = (0, 0, \dots)$
- ∞ , если \nexists РСЛОС, порождающего s .
- Длина минимального регистра сдвига, порождающего s .

Обозначим $L(s)$.

Определение 4.5. Пусть s — последовательность бит. Тогда пусть

$$L_N = L(s_0, \dots, s_{N-1})$$

Последовательность L_1, L_2, \dots назовем профилем линейной сложности последовательности s .

Утверждение 4.2. Верны следующие утверждения

1. $j > i \implies L_j \geq L_i$
2. $L_N \leq \frac{N}{2} \implies L_{N+1} > L_N$
3. $L_{N+1} > L_N \implies L_N + L_{N+1} = N + 1$

Сам алгоритм базируется на этих трех утверждениях. [Можно его дописать сюда].

4.3 Порождение симплексного кода с помощью регистра сдвига

Определение 4.6. Рассмотрим C_m , $(2^m - 1, 2^m - m - 1)$ -код Хэмминга. Дуальный к нему код S_m является кодом Адамара с матрицей Сильвестра — симплексным кодом.

Замечание 4.1. S_m является циклическим кодом с проверочным многочленом

$$h(x) = 1 + h_1 x + \dots + h_{m-1} x^{m-1} + x^m$$

Тогда, вспоминая структуру проверочной матрицы циклического кода, можем записать условия на то, что $(s_0, \dots, s_{2^m-2}) \in S_m$:

$$\forall i \in \{0, \dots, 2^m - 2 - m\}: s_{i+m} = s_i + s_{i+1} h_{m-1} + \dots + s_{i+m-1} h_1$$

Тогда каждое кодовое слово $s \in S_m$ порождается регистром сдвига с характеристическим многочленом $h(x)$ и начальными входами s_0, \dots, s_{m-1} .

Глава 5

Булевы функции и их свойства

5.1 Определения. Алгебраическая нормальная форма

5.1.1 Алгебраическая нормальная форма

Определение 5.1. Функция $f : \{0, 1\}^m \rightarrow \{0, 1\}$ называется булевой функцией

Мы поймем, что *любую* булеву функцию можно представить в виде многочлена от m переменных в \mathbb{F}_2 и даже выведем явную формулу.

С помощью таблицы истинности можно легко заметить, что для любой булевой функции f верно

$$f(v_1, \dots, v_m) = \bigvee_{i_1, \dots, i_m \in \{0, 1\}} f(i_1, \dots, i_m)(w_1^{i_1} \wedge \dots \wedge w_m^{i_m})$$

Где $w_i^1 = v_i$ и $w_i^0 = \neg v_i$.

Это просто функция в дизъюнктивной нормальной форме.

Определение 5.2. Будем обозначать $x \leq y$ для $x, y \in \{0, 1\}^n$, если $x_i \leq y_i$ для всех $i \in \{1, \dots, n\}$

Теорема 5.1. Любая булева функция f может быть записана как

$$f(v_1, \dots, v_m) = \sum_{a \in \{0, 1\}^m} g(a) v_1^{a_1} \dots v_m^{a_m}$$

$$\text{где } g(a) = \sum_{b \leq a} f(b_1, \dots, b_m)$$

Здесь и далее, если не указано иное, все суммы в \mathbb{F}_2

Доказательство. Зафиксируем набор значений v_1, \dots, v_m и проверим, что получается то, что нужно. Пусть $A = \{i : v_i = 1\} = \{\alpha_1, \dots, \alpha_k\}$; $B = \{i : v_i = 0\}$.

Во-первых, можно выбросить слагаемые, где $a_i = 1$ для $i \in B$, так как они обращаются в ноль.

$$\sum_{a \leq \mathbf{1}_A} g(a) v_1^{a_1} \dots v_m^{a_m} = \sum_{a \leq \mathbf{1}_A} g(a)$$

Здесь $\mathbf{1}_A$ — характеристический вектор A , он равен (v_1, \dots, v_m) .

Так как во всех остальных слагаемых $v_1^{a_1} \dots v_m^{a_m} = 1$. Тогда, подставляя $g(a)$, получаем

$$= \sum_{a \leq \mathbf{1}_A} \sum_{b \leq a} f(b_1, \dots, b_m)$$

Давайте поймем, сколько раз каждое слагаемое входит в сумму:

$$= \sum_{b \leq \mathbf{1}_A} \sum_{\mathbf{1}_A \geq a \geq b} f(b_1, \dots, b_m)$$

Осталось посчитать сколько бывает таких a . Легко видеть, что их количество равно $2^{w(a)-w(b)}$ и тогда все слагаемые, кроме $b = \mathbf{1}_A = (v_1, \dots, v_m)$ по четности обращаются в ноль

$$= f(v_1, v_2, \dots, v_m)$$

□

Определение 5.3. Представление

$$f(v_1, \dots, v_m) = \sum_{a \in \{0,1\}^m} g(a) v_1^{a_1} \dots v_m^{a_m}$$

называется *алгебраической нормальной формой* функции f .

5.1.2 Быстрое преобразование Мёбиуса

Определение 5.4. Пусть $f \in \{0,1\}^{2^n}$. Поставим вектору f в соответствие булеву функцию $\in \text{Map}(\{0,1\}^n, \{0,1\})$ следующим образом

$$f \mapsto \left(\underbrace{x}_{\in \{0,1\}^n} \mapsto f_x \right)$$

где f_x — компонента вектора f с номером, соответствующим двоичной записи x . Мы будем отождествлять вектор f и соответствующую ему функцию и записывать $f(x) = f_x$.

Отображение $\mu : \{0,1\}^{2^n} \rightarrow \{0,1\}^{2^n}$ называется *преобразованием Мёбиуса*, если выполнено:

$$f \mapsto g \iff \forall a: g(a) = \bigoplus_{b \leq a} f(b)$$

Вычисление преобразования Мёбиуса по определению требует 3^n операций (это количество пар $x, y \in \{0,1\}^n: x \leq y$). Но можно выполнить его оптимальнее, используя $2^n \cdot n$ операций.

Действительно, рассмотрим f и $g = \mu(f)$.

$$g(a_1, \dots, a_n) = \bigoplus_{b \leq a} f(b_1, \dots, b_n) = \bigoplus_{\substack{b \leq a \\ b_1=0}} f(b_1, \dots, b_n) \oplus \bigoplus_{\substack{b \leq a \\ b_1=1}} f(b_1, \dots, b_n)$$

Теперь разберем два случая: $a_1 = 0$ и $a_1 = 1$:

$$g(0, a_2, \dots, a_n) = \bigoplus_{\substack{b \leq a \\ b_1=0}} f(b_1, \dots, b_n)$$

в этом случае второе слагаемое обращается в ноль, поскольку $b_1 = 1 \implies b \not\leq a$.

$$g(1, a_2, \dots, a_n) = \bigoplus_{\substack{b \leq a \\ b_1=0}} f(b_1, \dots, b_n) \oplus \bigoplus_{\substack{b \leq a \\ b_1=1}} f(b_1, \dots, b_n)$$

Заметим, что теперь в обоих случаях условие $b \leq a \iff (b_2, \dots, b_n) \leq (a_2, \dots, a_n)$. Это дает нам возможность рассмотреть функции $f_0(a') = f(0, a'_1, \dots, a'_{n-1})$ и $f_1(a') = f(1, a'_1, \dots, a'_{n-1})$ и, используя прошлые рассуждения, записать

$$g(0, a_2, \dots, a_n) = \mu(f_0)(a_2, \dots, a_n)$$

и

$$g(1, a_2, \dots, a_n) = \mu(f_0)(a_2, \dots, a_n) \oplus \mu(f_1)(a_2, \dots, a_n)$$

Таким образом, мы свели задачу нахождения преобразования Мёбиуса для вектора из $\{0,1\}^{2^n}$ к двум задачам нахождения преобразования Мёбиуса для вектора из $\{0,1\}^{2^{n-1}}$ тогда время работы нашего алгоритма равно $T(n) = 2^n + 2T(n-1)$. Из этого соотношения легко видеть, что $T(n) = 2^n \cdot n$.

5.2 Коды Рида-Маллера

Определение 5.5. Для произвольного $r \in \{0, \dots, m\}$ двоичный код Рида-Маллера $\mathcal{R}(r, m)$ порядка r и длины 2^m определяется как

$$\text{Lin}\{v_{\alpha_1} \cdot \dots \cdot v_{\alpha_p} : p \leq r; 1 \leq \alpha_i \leq m\}$$

то есть линейная оболочка мономов степени $\leq r$ или, что то же самое, множество всех многочленов от m переменных над \mathbb{F}_2 степени не больше r .

Собственно кодами будут характеристические векторы этих многочленов.

Очевидно, что этот код является линейным. Значит, можно говорить о его размерности.

Замечание 5.1. Размерность $\mathcal{R}(r, m)$ равна $\sum_{k=0}^r C_m^k$

Доказательство. Из теоремы 5.1 все мономы линейно независимы, а количество мономов степени $\leq r$ равно $\sum_{k=0}^r C_m^k$ □

5.2.1 Взаимосвязь кодов Рида-Маллера разных порядков

Теорема 5.2.

$$\mathcal{R}(r+1, m+1) = \{|u|u+v| : u \in \mathcal{R}(r+1, m), v \in \mathcal{R}(r, m)\}$$

Здесь $|x|y|$ — конкатенация x и y

Лемма 5.1. Пусть $f(v_1, \dots, v_m)$ — булева функция с характеристическим вектором ϕ . Тогда характеристические векторы функций $g(v_1, \dots, v_{m+1}) = f(v_1, \dots, v_m)$ и $h(v_1, \dots, v_{m+1}) = v_{m+1}f(v_1, \dots, v_m)$ равны, соответственно $|\phi|\phi|$ и $|0|\phi|$

Доказательство. Здесь $m+1$ считается старшей степенью. Тогда левой части соответствуют те значения переменных где $v_{m+1} = 0$, а правой — те, где $v_{m+1} = 1$. Тогда в обеих частях характеристического вектора f будет вектор ϕ .левой части характеристического вектора h будет соответствовать тождественный 0, поскольку мы умножили на 0. □

Доказательство. Рассмотрим $f \in \mathcal{R}(r+1, m+1)$. Давайте запишем

$$f(v_1, \dots, v_{m+1}) = \underbrace{g(v_1, \dots, v_m)}_{\deg \leq r+1} + v_{m+1} \underbrace{h(v_1, \dots, v_m)}_{\deg \leq r}$$

Вспомним лемму и заметим, что характеристические векторы слагаемых этой формулы равны $|\mathbf{1}_g|\mathbf{1}_g|$ и $|0|\mathbf{1}_h|$ соответственно. Тогда характеристический вектор их суммы равен $|\mathbf{1}_g|\mathbf{1}_g + \mathbf{1}_h|$. □

Замечание 5.2. Похоже на формулу для биномиальных коэффициентов.

Теперь мы готовы к тому, чтобы найти расстояние между кодовыми словами в коде Рида-Маллера.

Теорема 5.3. Минимальное расстояние между словами в коде $\mathcal{R}(r, m)$ равно 2^{m-r}

Доказательство. Индукция по m . При $m = 0$ существует один код Рида Маллера: $\mathcal{R}(0, 0)$. Он состоит из слов 0 и 1, расстояние между ними равно 1.

Пусть для всех $m < m_0$ доказано, докажем для m_0 . Из прошлой теоремы $\mathcal{R}(r, m) = \mathcal{R}(r, m-1) + \mathcal{R}(r-1, m-1)$. Рассмотрим $a_1, a_2 \in \mathcal{R}(r, m)$. Они имеют вид $|u_1|u_1+v_1|$ и $|u_2|u_2+v_2|$ соответственно. Тогда

$$d(a_1, a_2) = d(u_1, u_2) + d(u_1+v_1, u_2+v_2) \geq d(u_1, u_2) + \underbrace{|d(u_1, u_2) - d(v_1, v_2)|}_{\geq 2^{m-r-1}} \geq 2^{m-r}$$

Поймём, что это неравенство действительно верно (здесь сумма вещественных чисел):

$$d(u_1+v_1, u_2+v_2) = \sum_{i=1}^{2^m} d_i(u_1+v_1, u_2+v_2)$$

где $d_i(x, y) = 1$, если i -е символы x и y различаются и 0, если совпадают. Теперь разбором случаев можно доказать, что $d_i(a+b, c+d) \leq |d_i(a, c) - d_i(b, d)|$:

| a | b | c | d | $d_i(a+b, c+d)$ | $d_i(a, c)$ | $d_i(b, d)$ | $ \dots $ |
|---|---|---|---|-----------------|-------------|-------------|-----------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

Здесь можно считать $a = b = 0$, так как иначе можно перейти к $a \rightarrow a + a$; $b \rightarrow b + b$; $c \rightarrow c + a$; $d \rightarrow d + b$, не изменив обе части формулы и обратив a, b в ноль. Таким образом можем записать

$$\sum_{i=1}^{2^m} d_i(u_1 + v_1, u_2 + v_2) \leq \sum_{i=1}^{2^m} |d_i(u_1, u_2) - d_i(v_1, v_2)| \leq \left| \sum_{i=1}^{2^m} (d_i(u_1, u_2) - d_i(v_1, v_2)) \right| = |d(u_1, u_2) - d(v_1, v_2)|$$

Теперь нужно разобрать два случая:

- $d(u_1, u_2) \geq d(v_1, v_2)$. Тогда $d(a_1, a_2) \geq 2^{m-r} + |\dots| \geq 2^{m-r}$
- $d(v_1, v_2) > d(u_1, u_2)$. Тогда $d(a_1, a_2) \geq d(u_1, u_2) + d(v_1, v_2) - d(u_1, u_2) = d(v_1, v_2) \geq 2^{m-r}$

□

5.2.2 Выколотые коды Рида-Маллера

Определение 5.6. Для произвольного $r \in \{0, \dots, m-1\}$ **выколотый** двоичный код Рида-Маллера $\mathcal{R}^*(r, m)$ порядка r и длины 2^m определяется как

$$\{x_1 x_2 \dots x_{2^m-1} : x \in \mathcal{R}(r, m)\}$$

то есть, получается из $\mathcal{R}(r, m)$ вычеркиванием элемента вектора, соответствующего $v_1 = \dots = v_m = 0$

Очевидно, что $\mathcal{R}^*(r, m)$ имеет длину $2^m - 1$, минимальное расстояние $2^{m-r} - 1$.

Утверждение 5.1. Для $r < m$ верно $\dim(\mathcal{R}^*(r, m)) = \sum_{k=0}^r C_m^k$

Доказательство. То есть, нужно доказать, что размерность равна размерности кода до выкалывания. Заметим, что по лемме о рандомизации, в каждой строке порождающей матрицы четное количество единиц (так как $r < m$ и строки, соответствующей $v_1 \dots v_m$, где только одна единица, в матрице нет).

Тогда сложим все столбцы, кроме первого, и получим столбец вида $(1, 0, \dots, 0)^T$.

Таким образом, размерность линейной оболочки всех столбцов $\mathcal{R}(r, m)$ равна размерности линейной оболочки всех столбцов, кроме первого, то есть $\dim(\mathcal{R}^*(r, m))$ □

5.3 Декодирование кода $\mathcal{R}(1, m)$

Рассмотрим на примере $m = 3$. Порождающая матрица $\mathcal{R}(1, m)$ будет иметь размер $(m+1) \times 2^m$ и будет состоять из векторов $\mathbf{1}, \mathbf{1}_{x_1}, \mathbf{1}_{x_2}, \mathbf{1}_{x_3}$:

$$G = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Кодирование, как и в любом линейном коде — домножение кодируемой строки на порождающую матрицу.

Все возможные 16 кодов получаются линейными комбинациями строк матрицы:

$$A = \mathcal{R}(1, 3) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Закодируем какое-нибудь слово. Например $y = (1, 0, 0, 1)$. $z = yG = (1, 1, 1, 1, 0, 0, 0, 0)$. Кодовое расстояние равно $2^{3-1} = 4$. Поэтому код способен исправить только одну ошибку. Тогда пусть $z' = (0, 1, 1, 1, 0, 0, 0, 0)$. Преобразуем матрицу A так, чтобы можно посчитать расстояния от декодируемого вектора до всех кодовых слов. $H_{ij} = 2A_{ij} - 1$. Можно заметить, что H — это две матрицы Адамара, поставленные друг на друга. Преобразуем z' тем же образом: $z''_i = 2z'_i - 1$.

Рассмотрим $H(z'')^T$. i -й компонент этого вектора равен $(2^m - d(H_i, z')) - d(H_i, z') = 2^m - 2d(H_i, z')$. Тогда вектор с минимальным расстоянием соответствует максимуму среди компонент $H(z'')^T$.

$$H = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

Тогда

$$H(z'')^T = (2, -2, 2, -2, 2, -2, 2, -2, -6, 6, 2, -2, 2, -2, 2, -2)^T$$

Максимальная компонента соответствует 10-й строке матрицы, а это и есть вектор $(1, 1, 1, 1, 0, 0, 0, 0)$, который мы шифровали.