Код Рида-Маллера

Илья Коннов

Факультет компьютерных наук

Высшая Школа Экономики

12 марта 2022 г.

Введение

Код описан Дэвидом Маллером (автор идеи) и Ирвингом Ридом (автор метода декодирования) в сентябре 1954 года.
Обозначается как $\mathrm{RM}(r, m)$, где r — ранг, а 2^m — длина кода. Кодирует сообщения длиной $k = \sum_{i=0}^{r} C_m^i$ при помощи 2^m бит.
Традиционно, считается что коды бинарные и работают над битами, т.е. $\mathbb{F}_2$.
Соглашение: сложение векторов $u, v \in \mathbb{F}_2^n$ будем обозначать как $u \oplus v = (u_1 + v_1, u_2 + v_2, ..., u_n + v_n)$.

Булевы функции и многочлен Жегалкина

Всякую булеву функцию можно записать при помощи таблицы истинности:

x	y	$f(x, y)$
0	0	1
0	1	0
1	0	0
1	1	0

Или при помощи многочлена Жегалкина:

$$f(x, y) = xy + x + y + 1$$

Многочлены Жегалкина

В общем случае, многочлены будут иметь следующий вид:

$$f(x_1, x_2, \dots, x_m) = \sum_{S \subseteq \{1,\dots,m\}} c_S \prod_{i \in S} x_i$$

Например, для $m = 2$: $f(x_1, x_2) = c_{12} \cdot x_{\{1\}} x_2 + c_{\{2\}} \cdot x_2 + c_{\{1\}} \cdot x_1 + c_{\varnothing} \cdot 1$
Всего $n = 2^m$ коэффициентов для описания каждой функции.

Функции небольшой степени

Рассмотрим функции, степень многочленов которых не больше r:

$$\{f(x_1, x_2, \ldots, x_m) \mid \deg f \le r\}$$

Каждую можно записать следующим образом:

$$f(x_1, x_2, \ldots, x_m) = \sum_{\substack{S \subseteq \{1,\ldots,m\} \\ |S| \le r}} c_S \prod_{i \in S} x_i$$

В каждом произведении используется не больше r переменных. Сколько тогда всего коэффициентов используется?

$$k = C_m^0 + C_m^1 + C_m^2 + \ldots + C_m^r = \sum_{i=0}^{r} C_m^i$$

Идея кодирования

Пусть каждое сообщение (длины k) — коэффициенты многочлена от m переменных степени не больше r.
Тогда мы можем его представить при помощи 2^m бит, подставив все возможные комбинации значений переменных.
Таким образом получим таблицу истинности, из которой позднее сможем восстановить исходный многочлен, а вместе с ним и сообщение.
Зафиксировав в таблице порядок строк, можно выделить **вектор значений**, который и будет кодом.

x	y	$f(x,y)$
0	0	1
0	1	0
1	0	0
1	1	0

$$\Longrightarrow \mathrm{Eval}(f) = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

Пример

- $r = 1$ (степень многочлена), $m = 2$ (переменных). Это $\mathrm{RM}(1, 2)$.
- Тогда наш многочлен: $f(x_1, x_2) = c_{\{2\}}x_2 + c_{\{1\}}x_1 + c_{\varnothing}$.
- Сообщение: `101`, тогда $f(x_1, x_2) = x_2 + 0 + 1$.
- Подставим всевозможные комбинации:

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	0

- Получили код: $\mathrm{Eval}(f) = $ `1100`.

Декодирование когда потерь нет

- Мы получили код: `1100`
- Представим таблицу истинности.

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	1
1	0	0
1	1	0

- Подстановками в $f(x_1, x_2) = c_2 x_2 + c_1 x_1 + c_0$ получим СЛАУ.

$$\begin{cases} & & & & c_0 & = & 1 \\ & & c_1 & + & c_0 & = & 1 \\ c_2 & + & & & c_0 & = & 0 \\ c_2 & + & c_1 & + & c_0 & = & 0 \end{cases}$$

- $c_{\{2\}} = 1, c_{\{1\}} = 0, c_\varnothing = 1$, исходное сообщение: `101`.

Коды 0-го порядка

Для случая $\mathrm{RM}(0, m)$ нужна функция от m аргументов, степени не выше 0.

- $f(x_1, x_2, ..., x_m) = 0$
- $g(x_1, x_2, ..., x_m) = 1$

Таблица истинности:

$$
2^m \left\{ \begin{array}{cccc|cc}
x_1 & x_2 & \dots & x_m & f(x_1, ..., x_m) & g(x_1, ..., x_m) \\
\hline
0 & 0 & \cdots & 0 & 0 & 1 \\
0 & 0 & \cdots & 1 & 0 & 1 \\
 & & \ddots & & \vdots & \vdots \\
1 & 1 & \cdots & 1 & 0 & 1
\end{array} \right.
$$

Вывод: это 2^m-кратное повторение символа

- Сообщение 0 даст код $\underbrace{\texttt{00...0}}_{2^m}$
- Сообщение 1 даст код $\underbrace{\texttt{11...1}}_{2^m}$

Коды m-го порядка

Есть m переменных, и мы рассматриваем многочлены $f \in \mathbb{F}_2[x_1, ..., x_m] : \deg f \leq m$, т.е. все возможные.
Для $\mathrm{RM}(m, m)$ мы используем все доступные коэффициенты многочлена для кодирования сообщения.
Тогда нет избыточности: $k = \sum_{i=0}^{m} C_m^i = 2^m = n$ – длина сообщения равна длине кода.

Чем меньше порядок кода r, тем больше избыточность.

Доказательство линейности

Пусть $C(x)$ кодирует сообщение $x \in \mathbb{F}_2^k$ в код $C(x) \in \mathbb{F}_2^m$.

$$C(x) = (p_x(a_i) \mid a_i \in \mathbb{F}_2^m)$$

где $p_x(a_i)$ — соответствующий сообщению x многочлен.
Причём p_x берёт в качестве своих коэффициентов биты из x. Поскольку многочлены степени не выше r образуют линейное пространство, то $p_{(x\oplus y)} = p_x + p_y$.
Тогда:

$$C(x \oplus y)_i = p_{(x\oplus y)}(a_i) = p_x(a_i) + p_y(a_i) = C(x)_i + C(y)_i$$

т.е. $\forall x, y \quad C(x \oplus y) = C(x) + C(y)$, ч.т.д.

Последствия линейности

1. Существует порождающая матрица G.

$$C(x) = x_{1\times k} G_{k\times n} = c_{1\times n}$$

2. Минимальное расстояние будет равно минимальному весу Хемминга среди всех кодов.

$$d = \min_{\substack{c\in C \\ c\neq 0}} w(c)$$

3. Корректирующая способность:

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

Конструкция Плоткина

Теорема

Для всякого кодового слова $c \in \mathrm{RM}(r, m)$ *можно найти* $u \in \mathrm{RM}(r, m-1)$ *и* $v \in \mathrm{RM}(r-1, m-1)$*, такие что* $c = (u \mid u+v)$*.*

Минимальное расстояние

Хотим найти минимальное расстояние для кода $\mathrm{RM}(r, m)$

$$d = \min_{c \in C, c \neq 0} w(c)$$

Предположим, что $d = 2^{m-r}$ и докажем по индукции.
База: $\mathrm{RM}(0, m)$ — единственный бит повторён 2^m раз. Очевидно, $w(\underbrace{11\ldots1}_{2^m}) = 2^m = 2^{m-0} \geq 2^{m-r}$.
Гипотеза: Если $v \in \mathrm{RM}(r-1, m-1)$, то $w(v) \geq 2^{m-r}$.
Шаг: Хотим доказать для $c \in \mathrm{RM}(r, m)$.

$$w(c) \overset{(1)}{=} w((u \mid u \oplus v)) \overset{(2)}{=} w(u) + w(u \oplus v) \geq$$
$$\overset{(3)}{\geq} w(u) + (w(v) - w(u)) = w(v) \overset{IH}{\geq} 2^{m-r} \blacksquare$$

Свойства и параметры

Для бинарного кода $\mathrm{RM}(r, m)$:

- $r \leq m$
- Длина кода: 2^m
- Длина сообщения: $k = \sum_{i=0}^{r} C_m^i$
- Минимальное расстояние: $d = 2^{m-r}$
- Корректирующая способность: $t = 2^{m-r-1} - 1$
- Существует порождающая матрица G для кодирования
- Проверочная матрица H совпадает с порождающей для $\mathrm{RM}(m-r-1, m)$

Возможные варианты

$m \backslash r$	0	1	2	3	4
1	$k=1$ $n=2$ $t=0$	$k=2$ $n=2$ $t=0$	—	—	—
2	$k=1$ $n=4$ $t=1$	$k=3$ $n=4$ $t=0$	$k=4$ $n=4$ $t=0$	—	—
3	$k=1$ $n=8$ $t=3$	$k=4$ $n=8$ $t=1$	$k=7$ $n=8$ $t=0$	$k=8$ $n=8$ $t=0$	—
4	$k=1$ $n=16$ $t=7$	$k=5$ $n=16$ $t=3$	$k=11$ $n=16$ $t=1$	$k=15$ $n=16$ $t=0$	$k=16$ $n=16$ $t=0$

Как линейный код

Этот код является линейным кодом, к нему применимы все обычные (и неэффективные методы):

- Перебор по всему пространству кодовых слов в поисках ближайшего.
- С использованием синдромов: $s = rH^T$.

Определения

1. Пусть $A \subseteq \{1, ..., m\}$ для $m \in \mathbb{N}$
2. Подпространство $V_A \subseteq \mathbb{F}_2^m$, которое обнуляет все v_i, если $i \notin A$: $V_A = \{v \in \mathbb{F}_2^m : v_i = 0 \ \forall i \notin A\}$
3. Аналогично для $V_{\bar{A}}$, где $\bar{A} = \{1, ..., m\} \setminus A$: $V_{\bar{A}} = \{v \in \mathbb{F}_2^m : v_i = 0 \ \forall i \in A\}$

Пример:

- Пусть $m = 3, A = \{1, 2\}$, тогда...
- $\mathbb{F}_2^m = \{\mathtt{000}, \mathtt{001}, \mathtt{010}, \mathtt{011}, \mathtt{100}, \mathtt{101}, \mathtt{110}, \mathtt{111}\}$
- $V_A = \{\mathtt{000}, \mathtt{010}, \mathtt{100}, \mathtt{110}\}$ $(v_3 = 0 \, \forall v)$
- $\bar{A} = \{1, 2, 3\} \setminus A = \{3\}$
- $V_{\bar{A}} = \{\mathtt{000}, \mathtt{001}\}$ $(v_1 = v_2 = 0 \, \forall v)$

Смежные классы

Если фиксировано $V_A \subseteq \mathbb{F}_2^m$, то для каждого $b \in \mathbb{F}_2^m$ существует смежный класс $V_A + b$:

$$(V_A + b) = \{v + b \mid v \in V_A\}$$

Утверждается, что если брать $b \in V_{\bar{A}}$, то полученные смежные классы будут все различны (и это будут все смежные классы).

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$:
$f(x_1, x_2) = u_{\{1,2\}}x_1x_2 + u_{\{2\}}x_2 + u_{\{1\}}x_1 + u_\varnothing$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \ldots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $c \mathrel{+}= \left(\sum_{z \in (V_A + b)} y_z\right) \bmod 2$

    $u_A \leftarrow \mathbf{1}\left[c \geq 2^{m-t-1}\right]$

  $y \mathrel{-}= \mathrm{Eval}\left(\sum_{\substack{A \subseteq \{1,\ldots,m\} \\ |A| = t}} u_A \prod_{i \in A} x_i\right)$

На вход поступает бинарный вектор y длины 2^m. Это вектор значений функции, возможно с ошибками (но их не больше, чем $t = 2^{m-r-1} - 1$).

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$:
$f(x_1, x_2) = u_{\{1,2\}}x_1x_2 + u_{\{2\}}x_2 + u_{\{1\}}x_1 + u_\varnothing$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \dots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $$c \mathrel{+}= \left(\sum_{z \in (V_A + b)} y_z\right) \bmod 2$$

    $u_A \leftarrow \mathbf{1}\left[c \geq 2^{m-t-1}\right]$

  $$y \mathrel{-}= \mathrm{Eval}\left(\sum_{\substack{A \subseteq \{1,\dots,m\} \\ |A| = t}} u_A \prod_{i \in A} x_i\right)$$

Будем восстанавливать сначала коэффициенты u_A при старших степенях, потом поменьше и так пока не восстановим их все. Начинаем с $t = r$.

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$: $f(x_1, x_2) = u_{\{1,2\}} x_1 x_2 + u_{\{2\}} x_2 + u_{\{1\}} x_1 + u_{\varnothing}$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \ldots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $c \mathrel{+}= \left(\sum\limits_{z \in (V_A + b)} y_z \right) \bmod 2$

    $u_A \leftarrow \mathbf{1}\left[c \geq 2^{m-t-1}\right]$

  $y \mathrel{-}= \mathrm{Eval}\left(\sum\limits_{\substack{A \subseteq \{1, \ldots, m\} \\ |A| = t}} u_A \prod_{i \in A} x_i \right)$

Хотим восстановить все коэффициенты при мономах степени t. Для этого перебираем все A, $|A| = t$ и для каждого восстанавливаем коэффициент u_A при $x_{A_1} x_{A_2} \ldots x_{A_t}$.

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$: $f(x_1, x_2) = u_{\{1,2\}}x_1x_2 + u_{\{2\}}x_2 + u_{\{1\}}x_1 + u_{\varnothing}$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \ldots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $$c \mathrel{+}= \left(\sum_{z \in (V_A + b)} y_z\right) \bmod 2$$

    $u_A \leftarrow \mathbf{1}\left[c \geq 2^{m-t-1}\right]$

  $$y \mathrel{-}= \mathrm{Eval}\left(\sum_{\substack{A \subseteq \{1,\ldots,m\} \\ |A| = t}} u_A \prod_{i \in A} x_i\right)$$

Чтобы восстановить коэффициент, нужно перебрать все смежные классы вида $(V_A + b)$:

$$V_A = \{v \in \mathbb{F}_2^m : v_i = 0\ \forall i \notin A\}$$

$$b \in \{v \in \mathbb{F}_2^m : v_i = 0\ \forall i \in A\}$$

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$: $f(x_1, x_2) = u_{\{1,2\}}x_1x_2 + u_{\{2\}}x_2 + u_{\{1\}}x_1 + u_{\varnothing}$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \ldots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $c \mathrel{+}= \left(\sum\limits_{z \in (V_A + b)} y_z\right) \mod 2$

    $u_A \leftarrow \mathbf{1}\,[c \geq 2^{m-t-1}]$

  $y \mathrel{-}= \mathrm{Eval}\left(\sum\limits_{\substack{A \subseteq \{1,\ldots,m\} \\ |A| = t}} u_A \prod_{i \in A} x_i\right)$

Считаем количество (c) смежных классов, в которых $\sum\limits_{z \in (V_A + b)} y_z = 1 \pmod 2$. Пороговое значение (2^{m-t-1}) здесь — половина от числа смежных классов. Таким образом, если большинство сумм дало 1, то $u_A = 1$, иначе $u_A = 0$.

Алгоритм Рида для кода $\mathrm{RM}(r, m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r, m)$. Для $\mathrm{RM}(2, 2)$: $f(x_1, x_2) = u_{\{1,2\}} x_1 x_2 + u_{\{2\}} x_2 + u_{\{1\}} x_1 + u_{\varnothing}$.

Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$

for $t \leftarrow r$ **to** 0

  foreach $A \subseteq \{1, \dots, m\}$ *with* $|A| = t$

    $c = 0$

    foreach $b \in V_{\bar{A}}$

      $c \mathrel{+}= \left(\sum_{z \in (V_A + b)} y_z \right) \bmod 2$

    $u_A \leftarrow \mathbf{1}\left[c \geq 2^{m-t-1}\right]$

  $y \mathrel{-}= \mathrm{Eval}\left(\sum_{\substack{A \subseteq \{1, \dots, m\} \\ |A| = t}} u_A \prod_{i \in A} x_i \right)$

Затем мы вычитаем из y (вектор значений функции) всё найденное на этой итерации, после чего переходим к мономам меньшей степени. Повторять до восстановления всех коэффициентов.

Пример

Ранее: `101` кодируется как `1100` при помощи $\mathrm{RM}(1,2)$

$$\texttt{101} \rightsquigarrow (f(x_1,x_2) = x_1 + 1) \rightsquigarrow \begin{array}{|cc|c|} x_1 & x_2 & f \\ \hline 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{array} \rightsquigarrow \begin{array}{rcl} y_{\texttt{00}} & = & 1 \\ y_{\texttt{01}} & = & 1 \\ y_{\texttt{10}} & = & 0 \\ y_{\texttt{11}} & = & 0 \end{array} \rightsquigarrow \texttt{1100}$$

Пример

Ранее: `101` кодируется как `1100` при помощи $\mathrm{RM}(1,2)$
Положим $y_{\mathtt{00}} = 1, y_{\mathtt{01}} = 1, y_{\mathtt{10}} = 0, y_{\mathtt{11}} = 0$
Здесь $m = 2$, значит $A \subseteq \{1,2\}$. Причём $r = 1$, т.е. $|A| \leq 1$.

Шаг 1/3: $t = 1, A = \{1\}$

- Здесь $V_A = \{\mathtt{00}, \mathtt{10}\}$, $V_{\bar{A}} = \{\mathtt{00}, \mathtt{01}\}$.
  Нужно рассмотреть два смежных класса.
- $(V_A + \mathtt{00}) = \{\mathtt{00}, \mathtt{10}\}$, сумма: $y_{\mathtt{00}} + y_{\mathtt{10}} = 1 + 0 = 1$
- $(V_A + \mathtt{01}) = \{\mathtt{01}, \mathtt{11}\}$, сумма: $y_{\mathtt{01}} + y_{\mathtt{11}} = 1 + 0 = 1$
- Итого: $u_A = u_{\{1\}} = 1$

Пример

Ранее: `101` кодируется как `1100` при помощи $\mathrm{RM}(1,2)$
Положим $y_{\texttt{00}} = 1, y_{\texttt{01}} = 1, y_{\texttt{10}} = 0, y_{\texttt{11}} = 0$
Здесь $m = 2$, значит $A \subseteq \{1,2\}$. Причём $r = 1$, т.е. $|A| \leq 1$.

Шаг 2/3: $t = 1, A = \{2\}$

- Здесь $V_A = \{\texttt{00}, \texttt{01}\}$, $V_{\bar{A}} = \{\texttt{00}, \texttt{10}\}$.
  Нужно рассмотреть два смежных класса
- $(V_A + \texttt{00}) = \{\texttt{00}, \texttt{01}\}$, сумма: $y_{\texttt{00}} + y_{\texttt{01}} = 1 + 1 = 0$
- $(V_A + \texttt{10}) = \{\texttt{10}, \texttt{11}\}$, сумма: $y_{\texttt{10}} + y_{\texttt{11}} = 0 + 0 = 0$
- Итого: $u_A = u_{\{2\}} = 0$

Пример

Ранее: `101` кодируется как `1100` при помощи $\mathrm{RM}(1,2)$
Положим $y_{\mathtt{00}} = 1, y_{\mathtt{01}} = 1, y_{\mathtt{10}} = 0, y_{\mathtt{11}} = 0$
Здесь $m = 2$, значит $A \subseteq \{1,2\}$. Причём $r = 1$, т.е. $|A| \leq 1$.

Перед переходом к $t = 0$, нужно вычесть из y вектор значений следующей функции:

$$g(x_1, x_2) = u_{\{2\}}x_2 + u_{\{1\}}x_1 = 0x_2 + 1x_1 = x_1$$

Вычислим $\mathrm{Eval}(g)$:

x_1	x_2	$g(x_1, x_2)$
0	0	0
0	1	0
1	0	1
1	1	1

Тогда $y \leftarrow y - \mathrm{Eval}(g) =$ `1100` $\oplus$ `0011` $=$ `1111`.

Продолжение примера: $t = 0$

Теперь $y_{\mathtt{00}} = 1, y_{\mathtt{01}} = 1, y_{\mathtt{10}} = 1, y_{\mathtt{11}} = 1$

Шаг 3/3: $t = 0, A = \varnothing$

- Здесь $V_A = \{\mathtt{00}\}$, но $V_{\bar{A}} = \{\mathtt{00}, \mathtt{01}, \mathtt{10}, \mathtt{11}\}$.
  Нужно рассмотреть **четыре** смежных класса.
- $(V_A + \mathtt{00}) = \{\mathtt{00}\}$, сумма: $y_{\mathtt{00}} = 1$
- $(V_A + \mathtt{01}) = \{\mathtt{01}\}$, сумма: $y_{\mathtt{01}} = 1$
- $(V_A + \mathtt{10}) = \{\mathtt{10}\}$, сумма: $y_{\mathtt{10}} = 1$
- $(V_A + \mathtt{11}) = \{\mathtt{11}\}$, сумма: $y_{\mathtt{11}} = 1$
- Итого: $u_A = u_{\varnothing} = 1$

Продолжение примера: $t = 0$

Теперь $y_{\texttt{00}} = 1, y_{\texttt{01}} = 1, y_{\texttt{10}} = 1, y_{\texttt{11}} = 1$

Получили $u_{\{2\}} = 1, u_{\{1\}} = 0, u_{\varnothing} = 1$.
Это значит, что исходный многочлен был таков:

$$f(x_1, x_2) = u_{\{2\}}x_2 + u_{\{1\}}x_1 + u_{\varnothing} = \color{red}{x_2 + 1},$$

а исходное сообщение: `101`, как и ожидалось.

Время работы

Утверждается, что время работы алгоритма — $O(n \log^r n)$, где $n = 2^m$ — длина кода.

Домашнее задание

Вариант 1

1. Закодировать сообщение: `1001`.
2. Декодировать код, если ошибок нет: `1010`, использовался $\mathrm{RM}(1,2)$.
3. Декодировать код, полученный с ошибками: `1101 1010`, использовался $\mathrm{RM}(1,3)$

Вариант 2

1. Закодировать сообщение: `0101`.
2. Декодировать код, если ошибок нет: `0110`, использовался $\mathrm{RM}(1,2)$.
3. Декодировать код, полученный с ошибками: `1111 0100`, использовался $\mathrm{RM}(1,3)$