Вопросы к экзамену

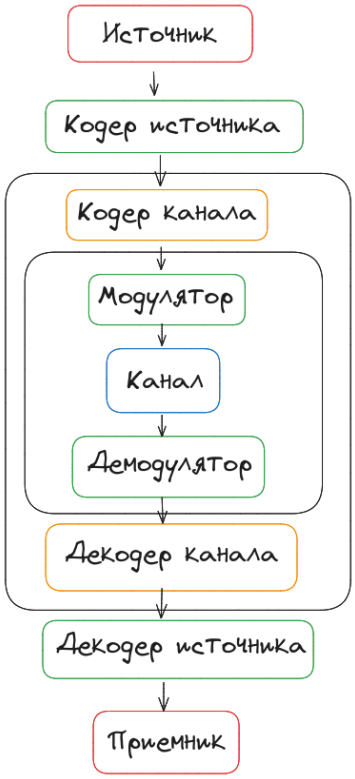
Ilya Yaroshevskiy

5 января

Содержание

1 Введение

1.1 Структура систем передачи информации



Определение. Код – множество допустимых последовательностей символов алфавита X, как конечных так и бесконечных

Определение. Кодер – устройство, реализующее отображение информационных последовательностей символов алфавита B в кодовые

Определение. Декодер – устройство, восстанавливающее по принятой последовательности символов *наиболее вероятную* соответствующую ей кодовую (или информационную) последовательность

1.2 Простейшие методы модуляции

Определение. Передаваемый сигнал равен

$$x(t) = \sum_i S_{x_i}(t - iT)$$

, где x_i – передаваемые символы, T – продолжительность символьного интервала

Пример. M-ичная амплитудно-импульсная модуляция

$$S_i(t) = \alpha(2i + 1 - M)g(t)\sin(2\pi ft)$$

, где $g(t)$ – сигнальный импульс (например, единичный импульс продолжительностью T), f – несущая частота, α – коэффицент, определяющий энергию передаваемого сигнала

Пример. Модель канала в непрервном времени $y(t) = x(t) + \eta(t)$

Пример. Модель канала в дискретном времени $y_i = (2x_i + 1 - M) + \eta_i$

Определение. $\eta_i \sim \mathcal{N}(0, \sigma^2)$ – канал с **аддитивным белым гауссовским шумом**

1.3 Критерии идеального наблюдателя и максимума правдоподобия

Замечание. Приемник наблюдает на выходе канала вектор $y = \begin{pmatrix} y_0 & \dots & y_{n-1} \end{pmatrix}$.
Канал характеризуется условным распределением $p_{Y|X}(y|x)$, где X, Y – случайные величины, соответствующие векторам переданных и принятых символов. Если выход канала – непрерывная случайная величина, $p_{Y|X}(y|x)$ – условная плотность вероятности. Приемник реализует некоторое разбиение векторного пространства на решающие области $R_x : y \in R_x \implies \hat{x} = x$

Определение. Вероятность ошибки

$$P_e = \int_{\mathbb{R}^N} p_e(y) p_Y(y) dy = \sum_x \int_{R_x} p_e(y) p_Y(y) dy =$$

$$= \sum_x \int_{R_x} (1 - p_{X|Y}(x|y)) p_Y(y) dy = 1 - \sum_x \int_{R_x} p_{X|Y}(x|y) p_Y(y) dy$$

Определение. Критерий максимума апостериорной вероятности (**критерий идеального наблюдателя**)

$$R_x = \{y | p_{X|Y}(x|y) > p_{X|Y}(x'|y), x' \neq x\} = \{y | P_X(x) p_{Y|X}(y|x) > P_X(x') p_{Y|X}(y|x'), x' \neq x\}$$

Определение. Критерий максимума правдоподобия

$$R_x = \{y | p_{Y|X}(y|x) > p_{Y|X}(y|x'), x' \neq x\}$$

1.4 Вероятность ошибки сигналов 2-AM в случае канала с АБГШ

Пример. 2-ичная амплитудно-импульсная модуляция (2-AM). Пусть $y_i = \alpha(2x_i - 1) + \eta_i, \eta_i \sim \mathcal{N}(0, \sigma^2), x_i \in \{0, 1\}$. Тогда:

$$p_{Y|X}(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - \alpha(2x-1))^2}{2\sigma^2}}$$

Применим критерий максимального правдоподобия:

$$R_0 = \{y | y < 0\}, R_1 = \{y | y \geq 0\}$$

Вычислим вероятность ошибки:

$$P_e = P_X(0) P\{Y \geq 0 | X = 0\} + P_X(1) P\{Y < 0 | X = 1\} = \dots = \int_{\frac{\alpha}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} dy = Q\left(\frac{\alpha}{\sigma}\right) =$$

$$= \frac{1}{2} erfc\left(\frac{\alpha}{\sqrt{2}\sigma}\right)$$

1.5 Отношение сигнал/шум на бит и на символ

Замечание. Значение сигнала это обычно уровень напряжения. Как мы знаем мощность $P = \frac{U^2}{R}$. Мы хотим минимизировать мощность, чтобы экономить электроэнергию. Мощность сигнала суть случайная величина с матожиданием, пропорциональным $E_S = \alpha^2$. Мощность белого шума не зависит от частоты и пропорциональна $\sigma^2 = \frac{N_0}{2}$. Если же шум зависит от частоты, то он называется розовым или голубым.

Соотношение мощностей сигнал/шум на символ это $\frac{E_S}{N_0}$, обычно измеряемое в децибелах, т.е. $10 \log_{10} \frac{E_S}{N_0}$. Однако нас интересуют не символы, а биты и тогда соотношение сигнал/шум на бит это $\frac{E_S}{RN_0}$, где R – количество бит информации, представленных одним символом.

2 Блоковые коды

2.1 Блоковые коды и их параметры

Замечание. Блоковые коды преобразуют блок из k символов в блок из n символов. Преобразование отдельных блоков выполняется независимо

Замечание. n – длина кода $\mathcal{C}$. Для исправления ошибок требуется, чтобы не все $|X|^n$ последовательностей были кодовыми словами. Мощность кода (число различных кодовых слов) $M = |C|$.

Замечание. Скорость кода: $R = \frac{\log_{|X|} M}{n}$.

Определение. Минимальным расстоянием кода называется минимальное расстояние Хемминга между его различными кодовыми словами

Утверждение. *Код с минимальныи расстоянием Хемминга d способен исправить* $\left\lfloor \frac{d-1}{2} \right\rfloor$

2.2 Критерии декодирования и метрики

Определение. Критерий минимального расстояния $X = Y$. Декодер ищет кодово слово

$$c = \text{argmin}_{c \in \mathcal{C}} d(c, y)$$

Определение. Алгоритм называется алгоритмом полного декодирования по критерию K, если он способен найти решение соответствующуй оптимизационной задачи для любого возможного принятого сигнала

Определение. Функция $d(x, y)$ называется метрикой, если:

- $d(x, y) \geq 0, d(x, y) = 0 \Leftrightarrow x = y$
- $d(x, y) = d(y, x)$
- Неравенстно треугольника $d(x, y) + d(y, z) \geq d(x, z)$

Определение. Метрическое пространство – множество X с определенной на нем метрикой

Пример. Расстояние Хемминга $d_H(x, y) = |\{i | x_i \neq y_i\}|$. Двоичный симметричный канал ($p < 0.5, X = Y = \{0, 1\}$:

$$\hat{c} = \text{argmax}_{x \in \emptyset} \prod_{i=1}^{n} P\{y_i | c_i\} = \cdots = \text{argmin}_{c \in \mathcal{C}} \sum_{i=1}^{n} a|y_i - c_i|$$

Пример. Расстояние Евклида $d_E(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$. Аддитивный Гауссовский канал с амплитудно-импульсной модуляцией ($Y = \mathbb{R}^n$).

Пример. Расстояние Ли ($A = GF(q)^n$) : $d_L(x, y) = \sum_{i=1} \min(|x_i - y_i|, q - |x_i - y_i|)$. Аддитивный Гауссовский канал с q-ичной фазовой модуляцией

Пример. Ранговое расстояние ($A = GF(q)^{n \times m}$) : $d_R(x, y) = \text{rank}(x - y)$. Сетевые коды

2.3 Границы Хемминга и Варшамова-Гилберта

Теорема 2.1 (Граница Хемминга)**.** Для любого q-ичного кода с минимальным расстоянием $d = 2t + 1$ число кодовых слов удовлетворяет

$$A_q(n, d) \leq \frac{q^n}{\sum_{i=0}^{t} C_n^i (q-1)^i}$$

Теорема 2.2 (Граница Варшамова-Гильберта)**.** Существует q-ичный код длины n с минимальными расстоянием d, число слов которого удовлетворяет

$$A_q(n, d) \geq \frac{q^n}{\sum_{i=0}^{d-1} C_n^i (q-1)^i}$$

3 Линейные коды

3.1 Линейные коды

Определение. Линейным (n,k) **(длины** n **размерностью** k**) кодом** C называется k-мерное линейное подпространство n-мерного линейного пространства над полем $GF(q)$.

Замечание. Число кодовых слов равно q^k

Определение. Порождающая $k \times n$ матрица полного ранга $G : C = \{y = xG | x \in GF(q)^k\}$

Определение. Проверочная матрица $r \times n$, $H : C = \{y \in GF(q)^n | yH^T = 0\}$, $r \geq n - k = \text{rank}(H)$

$$GH^T = 0$$

Замечание. С помощью линейных операций над строками и перестановок столбцов порождающая матрица может быть приведена к виду $G = (I|A)$.
NB Вообще говоря перестановкой столбцов получается порождающая матрица другого кода, поэтому перестановка столбцов в порождающей матрице должна быть согласована с перестановкой в проверочной

Замечание. Систематическое кодирование $xG = (x|xA)$ – информационный вектор является подвектором кодого слова. Применение систематического кодирования упрощает декодирование

$$H = (A^T| - I)$$

Утверждение. *Минимальное расстояние линейного блокового кода* $\mathcal{C}$ *равно* $d = \min\limits_{\substack{c' \neq c'' \\ c', c'' \in \mathcal{C}}} d(c', c'') = \min_{c \in \mathcal{C} \setminus \{0\}} wt(c)$*, где* $wt(c)$ *– вес вектора (количество единиц)*

Утверждение. *Если* H *– проверочная матица кода длины* n*, то код имеет размерность* $n - r \Leftrightarrow$ *существуют* r *линейно независимых столбцов матрицы* H*, а любые* $r + 1$ *столбцов линейно зависимы*

Утверждение. *Если* H *– проверочная матрица кода длины* n*, то код имеет минимальное расстояние* $d \Leftrightarrow$ *любые* $1, 2, \ldots, d - 1$ *столбцов* H *линейно независимы, но существуют* d *линейно зависимых столбцов матрицы* H

Определение. Коды с $n - k = d - 1$ называются **разделимыми** кодами с максимальным достижимым расстоянием

3.2 Границы Синглтона, Варшамова-Гилберта и Грайсмера

Утверждение. *Граница Синглтона (верхняя): для любого* (n, k, d) *линейного кода* $n - k \geq d - 1$

Теорема 3.1 (Граница Варшамова-Гильберта для линейных кодов)**.** Если выполняется $q^r > \sum_{i=0}^{d-2} C_{n-1}^i (q-1)^i$, то существует линейный код над $GF(q)$ длины n с минимальными расстоянием не менее d и не более чем $r = n - k$ проверочными символами

Определение. $N(k, d)$ – минимальная длина двоичного линейного кода размерности k с минмальными расстоянием d.

Теорема 3.2 (Граница Грайсмера)**.** $N(k, d) \geq d + N(k - 1, \lceil d/w \rceil)$

3.3 Вероятность ошибки декодирования и необнаружения ошибки

Определение. Весовой спектр кода $A_i = |\{c \in C | wt(c) = i\}|$.

Определение. Пусть есть подгруппа некоторой группы G, $G' \subset G$. Возьмем элемент $a \in G$, то **смежным классом** подгруппы G', называется $aG' = \{a \cdot x | x \in G'\} \subseteq G$

Определение. Лидер смежного класса – минимальный по весу вектор.

Замечание. Рассмотрим двоичный симметричный канал с переходной вероятностью p. Вероятность не обнаружения ошибки:

$$P_{\text{undetect}} = P\{S = 0\} = \sum_{i=d}^{n} A_i p^i (1-p)^{n-i} \leq \sum_{i=d}^{n} C_n^i p^i (1-p)^{n-i}$$

Вероятность правильного декодирования. Вероятность того, что вектор ошибки является лидером смежного класса:

$$P_{\text{correct}} = \sum_{i=0}^{l} L_i p^i (1-p)^{n-i}$$

, где L_i – число лидеров смежных классов веса i, l – максимальный вес лидера смежного класса

4 Декодирование линейных кодов

4.1 Декодирование линейных кодов

В классичесической архитектуре предполагается что модулятор преобразует закодированные данные в сигнал, демодулятор оценивает символы кодовых слов, а декодер потом пытается исправить ошибки. Такой подход плох тем что теряется информация о надежности отдельных принятых символов. Сейчас как правило используют мягкое декодирование: демодулятор каким-то образом формирует информацию о надежности отдельных принятых символов, декодер при исправлении пытается учесть эту информацию.

Определение. Синдром принятого вектора $S = yH^T = xGH^T + eH^T = eH^T$ зависит только от вектора ошибки

4.2 Таблица стандартной расстановки

Замечание. Рассмотрим все возможные вектора e и выпишем соответсвующие синдромы. Отсортируем по весу все возможные вектора e, соответствующие каждому возможному значению синдрома (*стандартная расстановка*). В качестве решения задачи декодирования выбираем самый легкий вектор e, соответствующий вычисленному синдрому:

4.3 Код Хемминга

Определение. Выберем в качестве столбцов матрицы H все ненулевые двоичные векторы длины r:

- Длина кода $n = 2^r - 1$
- Размерность $k = n - r = 2^r - r - 1$
- Минимальное расстояние $d = 3$

Замечание. Если произошла только одна ошибка, то $S = eH^T$ будет равно какому-то столбцу матрицы H. Получается синдром – двоичное представление числа, которое является номером позиции в которой произошла ошибка.

4.4 Стирания

Замечание. Некоторые символы могут просто теряться. Стирания могут происходить одновременно с ошибками. Утверждается что (n, k, d) код может исправить любую комбинацию из t ошибок и v стираний, если $d \geq 2t + v + 1$. Стирание эквивалентно выкалыванию кода на v позиций $\implies$ минимальное расстояние уменьшается не более чем на v.

Замечание. Декодирование ошибок и стираний для кодов над $GF(2)$:

- Положить все стертые позиции равными 0, исправить ошибки в полученном векторе
- Положить все стертые позиции равными 1, исправить ошибки в полученном векторе
- Выбрать результат декодирования, ближайший к принятому вектору

5 Информационные совокупности

5.1 Декодирование по информационным совокупностям

Определение. Информационной совокупностью называется множество из k позиций в кодовом слове, значения которых однозначно определяют значения на остальных позициях кодового слова

Определение. Если $\gamma = \{j_1, \dots, j_k\}$ – ИС, то все прочие позиции $\{1, \dots, n\} \setminus \gamma$ образуют **проверочную совокупносить**

Утверждение. *Если $\gamma = \{j_1, \dots, j_k\}$ образует ИС, то матрица, составненная из столбцов $j_1, \dots, j_k$ порождающей матрицы, обратима*

Определение. $M(\gamma) = A^{-1}$

Утверждение. $G(\gamma) = M(\gamma)G$ – *породжающая матрица, содержащая единичную подматрицу на столбцах γ, где $M(\gamma)$ – подходящая обратимая матрица*

Замечание. $G = (A|B), G(\gamma) = (\overbrace{I}^{\gamma}|M(\gamma)B)$

Определение. ИС **свободна от ошибок**, если соответствующие позиции вектора e равны 0: $e(\gamma) = 0$

Замечание. Декодирование $y = xG + e$ по информационным совокупностям:

- (первоначальный кандидат) $c = 0$
- Выбрать ИС γ. Вычислить $c' = y(\gamma)G(\gamma)$
- Если $d(c', y) < d(c, y), c = c'$
- Перейти к следующей ИС. Если все ИС проверены, возвратить c.
- Не всякие k позиций образуют информационную совокупность

Теорема 5.1. Алгоритм декодирования по ИС обеспечивает полное декодирование по минимальному расстоянию

6 Дуальные коды

6.1 Дуальные коды

Определение. Пусть задан (n, k) код с проверочной матрицей H. **Дуальным** к нему называется $(n, n-k)$ код с порождающей матрицей H.

Замечание. Скалярное произведение кодового слова из дуального кода на слово из исходного кода равно 0.

Определение. Самодуальным называется код, совпадающий со своим дуальным

Утверждение. *Код с проверочной матрицей $H = (A|I)$ самодуален ттогда, когда A – квадратичная матрица, такая что $AA^T = -I$*

$$HH^T = AA^T + I$$

6.2 Весовой спектр кода

Определение. Спектром линейного кода называется последовательность $A_i, i = 0 \dots n$, где A_i равно числу кодовых слов веса i

6.3 TODO Тождество Мак-Вильямс и его доказательство

Определение. Весовая функция кода

$$W(x,y) = \sum_{i=0}^{n} A_i x^{n-i} y^i = \sum_{u \in C} x^{n-wt(u)} y^{wt(u)}$$

Теорема 6.1 (Мак-Вильямс для двоичных линейных кодов)**.** Весовая функция кода C связана с весовой функцией дуального к нему кода $C_\perp$ соотношением

$$W_{C_\perp} = \frac{1}{|C|} W_C(x+y, x-y)$$

7 Критерии мягкого декодирования

7.1 Критерии мягкого декодирования

Утверждение. *Декодирования кода C по критерию максимуму правдоподобия в канале с АБГШ эквивалентно декодированию по критерию минимального расстояния Евклида*

Замечание. Рассмотрим передачу по каналу с АБГШ символов $(-1)^{c_i}, c_i \in \{0,1\}$, т.е. $y_i = (-1)^{c_i} + \eta_i$

$$\text{argmin}_{c \in C} \sum_{i=0}^{n-1} (y_i - (-1)^{c_i})^2 = \text{argmin}_{c \in C} \sum_{i=0}^{n-1} (y_i^2 - 2(-1)^{c_i} y_i + (-1)^{2c_i}) = \text{argmax}_{c \in C} \sum_{i=0}^{n-1} (-1)^{c_i} y_i$$

Пусть $\hat{c}_i = \begin{cases} 0 & , y_i > 0 \\ 1 & , y_i \le 0 \end{cases}$ – жесткие решения

$$\text{argmax}_{c \in C} \sum_{i=0}^{n-1} (-1)^{c_i} y_i = \text{argmax}_{c \in C} \sum_{i=0}^{n-1} ((-1)^{c_i} y_i - (-1)^{\hat{c}_i} y_i) = \text{argmax}_{c \in C} \sum_{i: c_i \neq \hat{c}_i} -|y_i| = \text{argmin}_{c \in C} E(c,y)$$

, где $E(c,y) = \sum_{i: c_i \neq \hat{c}_i} |y_i|$ – *корреляционная невязка.* y_i может быть заменено на логарифмические отношения правдоподобия $L_i = \log \frac{P\{c_i=0|y_i\}}{P\{c_i=1|y_i\}}$

Замечание. Задача минимизации евклидового расстояния эквивалентна задаче максимизации корреляция и эквивалентная задаче минимизации корреляционной невязки

7.2 Метод порядковых статистик

Замечание. Рассмотрим передачу кодовых слов $(c_0, \ldots, c_{n-1})$ двоичного (n,k) кода с помощью символов 2-АМ по каналу без памяти. Пусть $(y_0, \ldots, y_{n-1})$ – соответствующие принятые символы.

Пример. $y_i = (-1)^{c_i} + \eta_i,\ \eta_i \sim N(0, \sigma^2)$

- Пусть $L_i = \log \frac{P\{c_i=0|y_i\}}{P\{c_i=1|y_i\}}$ – логарифмические отношения правдоподобия
- Пусть $\hat{c}_i = \begin{cases} 0 & , L_i > 0 \\ 1 & , L_i < 0 \end{cases}$ – жесткие решения

Вероятность ошибки в $\hat{c}_i$ убывает с увеличением $|L_i|$. Выберем информационную совокупность J кода, соответствующую наибольшим значениям $|L_i|$. Приведем порождающую матрицу кода к виду G_J с единичной подматрицей в столбцах J. С большой вероятностью число неверных жестких решений $\hat{c}_i, i \in J$, мало. Переберем все конфигурации ошибок e веса не более t на J и построим кодовые слова $c_e = (\hat{c}_J + e)G_J$. Выберем наиболее правдоподобное из полученных кодовых слов. Сложность $O(k^2 n + \sum_{i=0}^{t} i n C_k^i)$. При $t = d/4$ достигается вероятность ошибки, близкая к вероятности ошибки декодирования по максимуму правдоподобия

8 Решетки

Замечание. Рассмотрим двоичный (n, k, d) код C с порождающей матрицей G. Пусть $D(x, y)$ – функция расстояния Хемминга. Декодирвание векотора y по максимуму правдоподобия

$$\hat{c} = \text{argmin}_{c \in C} D(c, y) = \text{argmin}_{c \in C} \sum_{i=0}^{n-1} D(c_i, y_i)$$

Существует много кодовых слов содержащих одинаковые префиксы $(c_0, \ldots, c_a)$. Было бы неразумно пересчитывать $\sum_{i=0}^{a} D(c_i, y_i)$ для них каждый раз заново. Декодирование – задача поиска ближайшего кодового слова. Попытаемся сформулировать ее как задачу поиска кратчайшего пути на графе

Определение. Решеткой (trellis) называется граф, обладающий следующими свойствами

- Вершины разбиты на непересекающиеся подмножества (уровни или ярусы)
- Нулевой и последний ярусы содержат по 1 узду (терминальные узлы)
- Грав направленный. Допускается движение только от уровня с меньшим номером к уровню с болшим нжомером. Стрелки при этом, как правило, не рисуют
- Ребрам графа приписаны метки, соответсвующие символам кодовых слов, а также метрики, называемые также весами или длинами. Длина пути равна сумме длин входящих в него ребер. Пример метрики ребра на ярусе i, помеченного c: $D(c, y_i)$

Замечание. Сопоставим пути в решетке между терминальными узлами кодовым словам. Тогда задача поиска кодового слова, минимизирующего некоторую аддитивную функцию функцию длины эквивалентна задаче поиска кратчайшего пути между терминальными узлами решетки

8.1 Минимальная решетка линейного блокового кода и способы ее построения

Определение. Профиль сложности решетки: $(\xi_0, \ldots, \xi_n), \xi_i = |V_i|$ – число узлов на i-м ярусе

Определение. Решетка называется **минимальной**, если профиль сложности $(\xi'_0, \ldots, \xi'_n)$ любой другой решетки удовлетворяет $\xi'_i \geq \xi_i, 0 \leq i \leq n$.

Замечание. Как построить минимальную решетку

Выпишем все кодовые слова $e_m = (c_{m,0}, \ldots, c_{m,n-1})$ рассмтриваемого кода. Для некоторого i определим прошлое $c^p_m = (c_{m,0}, \ldots, c_{m,i-1})$ и будущее $c^f_m = (c_{m,i}, \ldots, c_{m,n-1})$. В любой решетке пути, входящие в некоторый узел, имеют общее будущее, а пути, исходящие из одного узла имеют общее прошлое.

$F_i = \{c^f | \exists c^p : (c^p, c^f) \in C\}$ может быть единственным образом разбито на подмножество $F_i(c^p) = \{c^f | (c^p, c^f) \in C\}$. Сопоставим каждому такому подмножеству узлы на ярусе i. Узел $v \in V_i$ свяжем с узлом $v' \in V_{i+1}$, если для некоторого кодового слова прошлое, соответствующее v' является продолжением на 1 символ одной из последовательностей, ведущих в узле v. Пометим этим символом ребро (v, v').

8.1.1 По порождающей

Замечание. Для (n, k) линейного кода C для каждого $i, 0 \leq i \leq n$ прошлое и будущее также являются линейными кодами. Для простроения решетки удобно найти базисы (порождающие матрицы) этих кодов. Удобно сделать это сразу для всех i. Началом $b(x)$ вектора $(x_0, \ldots, x_{n-1})$ будем называть номер первого его ненулевого элемента. Концом $e(x)$ вектора $(x_0, \ldots, x_{n-1})$ будем называть номер последнего его ненулевого элемента. Элементы на позициях $b(x), \ldots, e(x) - 1$ будем называть активными.

Определение. Порождающая матрица называется **приведенной к минимальной спэновой форме** (МСФ), если все начала строк различны и все концы строк различны.

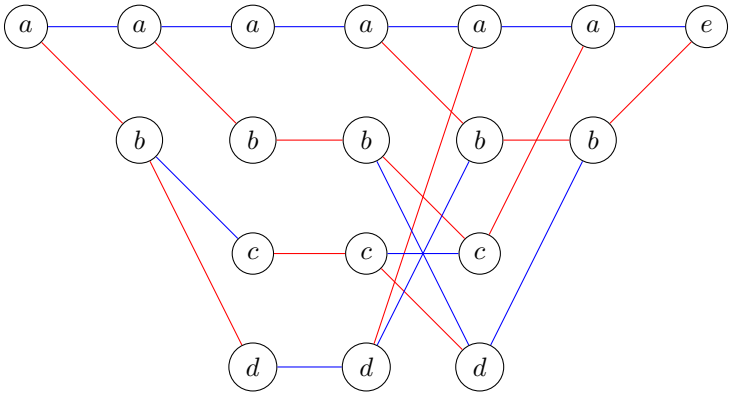
Замечание. Для определенности потребуем также, чтобы строки матрицы были упорядочены по возрастанию начал строк. Матрица может быть приведена к МСФ с помощью элементарных операций над строками

Пример. Порождающая матрица

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

Кодовые слова: $c_0 = (000000), c_1 = (110100), c_2 = (101010), c_3 = (011110), c_4 = (101101), c_5 = (011001), c_6 = (000111), c_7 = (110011)$

i	c^p	$F_i(c^p)$	v_i	v_{i-1}	$c_{i,v_i,v_{i-1}}$
0	$\varnothing$	$\{c_m \mid 0 \le m \le 7\}$	a	–	–
1	0	$00000, 11110, 11001, 00111$	a	a	0
	1	$10100, 01010, 01101, 10011$	b	a	1
2	00	$0000, 0111$	a	a	0
	01	$1110, 1001$	b	a	1
	10	$1010, 1101$	c	b	0
	11	$0100, 0011$	d	b	1
3	000	$000, 111$	a	a	0
	011	$110, 001$	b	b	1
	101	$010, 101$	c	c	1
	110	$100, 011$	d	d	0
4	$0000, 1101$	00	a	a, d	$0, 1$
	$0001, 1100$	11	b	a, d	$1, 0$
	$0111, 1010$	10	c	b, c	$1, 0$
	$0110, 1011$	01	d	a, c	$0, 1$
5	$00000, 11010, 10101, 01111$	0	a	a, c	$0, 1$
	$10110, 01100, 00011, 11001$	1	b	b, d	$1, 0$
6	$\{c_m \mid 0 \le m \le 7\}$	$\varnothing$	a	a, b	$0, 1$



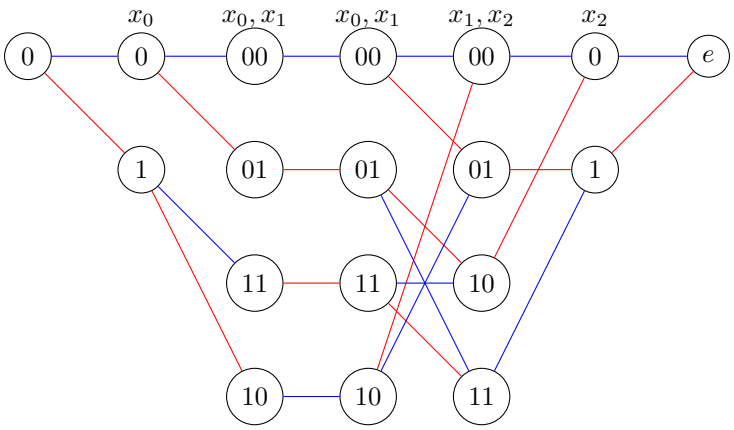
синие ребра – 0, красные – 1

Пример. Приведем порождающую матрицу к МСФ

$$G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \xrightarrow{(1)+(2),(1)+(3),(2)\leftrightarrow(3)} \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\xrightarrow{(2)+(1),(3)+(2)} \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

На ярусе i узлы нумеруются значениями информационных символов, соответсвующих строкам МСФ, активным в позиции i. Ребра помечаются линейными комбинациями активных элементов столбца порождающей матрицы. Полученная решетка минимальна



8.1.2 По проверочной

Замечание.

- Пусть дана проверочная матрица $H = (h_0, \ldots, h_{n-1})$.
- Пусть $S_0 = 0$
- Накопленный синдром $S(x_0, \ldots, x_{j-1}) = \sum_{i=0}^{j-1} h_i x_i$

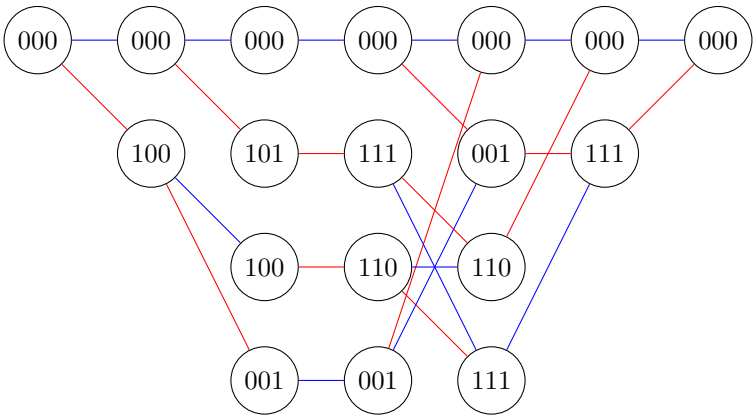
Будем нумеровать узлы v в решетке накопленными синдромами $S(v)^T$. Существует ребро, помеченное c, из v' на ярусе i в v на ярусе $i+1$, если $S(v) = S(v') + ch_i$. Оставим единственный конечный узел, соответсвующий нулевому синдрому (т.е. допустимым кодовым словам). Удалим нетерминальные узлы, из которых не выходят ребра. Полученная решетка минимальна.

Теорема 8.1. Решетка, построенная по проверочной матрице, минимальна

Теорема 8.2. Всякий код имеет минимальную решетку, все минимальные решетки совпадают с точностью до нумерации узлов каждого яруса

Пример.

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



8.2 Алгоритм Витерби

Program 1 Viterbi(y)

1: $M_{0,0} = 0$
2: **for** $i = 1, \ldots, n$ **do**
3: **for** $v \in V_i$ **do**
4: $M_{v',v} = M_{i-1,v'} + D(c[i, v', v], y_{i-1})$ {Для каждого входящего ребра (v', v) вычислить метрику его пути}
5: $M_{i,v} = \min M_{v',v}$ {Метрика узла}
6: $B_{i,v} = \text{argmin} M_{v',v}$ {Узел-предшественник}
7: **end for**
8: **end for**
9: $v = 0$
10: **for** $i = n, \ldots, 1$ **do**
11: $\hat{c}_{i-1} = c[B_{i,v}, v]$
12: $v = B_{i,v}$
13: **end for**
14: **return** $\hat{c}, M_{n,0}$

Замечание.

- $M_{i,v}$ – метрика узла v на ярусе i
- V_i – множества узлов на ярусе i
- $c[i, v', v]$ – метка ребра между узлами $v' \in V_{i-1}, v \in V_i$
- Число сложений не превосходит E (число ребер в решетке)
- Число сравнений не превосходит $E - V$, где V – число узлов

9 Декодирование с мягким выходом

9.1 Декодирование с мягким выходом

Замечание. Длинные коды можно построить, комбинируя короткие. Декодеры "составных"кодов могут быть построены из декодеров компонентных кодов. Взаимодействие декодеров может осуществляться путем обмена апостериорными вероятностями

$$p\{c_i = a|y_0^{n-1}\} = \sum_{c \in C_i(a)} p\{c|y_0^{n-1}\}$$

- $y_0^{n-1} = (y_0, \ldots, y_{n-1})$ – результат передачи кодовго слова кода C по каналу без памяти
- $C_i(a) = \{(c_0, \ldots, c_{n-1}) \in C | c_i = a\}$

Апостериорные логарифмические отношения правдоподобия для двоичных кодов

$$L_i = \ln \frac{p\{c_i = 0|y_0^{n-1}\}}{p\{c_i = 1|y_0^{n-1}\}}$$

Исходные данные ЛОПП символов кодового слова $L_i = \ln \frac{p(y_i|c_i=0)}{p(y_i|c_i=1)}$

Замечание. Такое декодирование называется декодированием с мягким входом и мягким выходом (soft input soft output, SISO)

9.2 Алгоритм БКЕР

$$L_i = \ln \frac{P\{c_i = 0|y_0^{n-1}\}}{P\{c_i = 1|y_0^{n-1}\}} = \ln \frac{\sum_{(s',s) \in S_0} \frac{p(s_i = s', s_{i+1} = s, y_0^{n-1})}{p(y_0^{n-1})}}{\sum_{(s',s) \in S_1} \frac{p(s_i = s', s_{i+1} = s, y_0^{n-1})}{p(y_0^{n-1})}}$$

где S_0 и S_1 – множества пар состояний $s' \in V_i, s \in V_{i+1}$, переход между которыми помечен соответственно 0 и 1, $p(y_0^{n-1})$ – совместная плотность распределения принятых сигналов, $p(s_i = s', s_{i+1} = s, y_0^{n-1}$ – совместная плотность распределения принятых сигналов и состояний кодера на ярусах i и $i+1$

Поведение кодера при обработке i-го символа определяется только его состоянием s' на предыдущем шаге; канал не имеет памяти

$$p(s_i = s', s_{i+1} = s, y_0^{n-1}) = p(s_i = s', y_0^{i-1}) p(s_{i+1} = s, y_i | s_i = s', y_0^{i-1}) p(y_{i+1}^{n-1} | s_{i+1} = s, s_i = s', y_0^i) =$$
$$= \underbrace{p(s_i = s', y_0^{i-1})}_{\alpha_i(s')} \underbrace{p(s_{i+1} = s, y_i | s_i = s')}_{\gamma_{i+1}(s',s)} \underbrace{p(y_{i+1}^{n-1} | s_{i+1} = s)}_{\beta_{i+1}(s)}$$

Из формулы Байеса:

$$\alpha_i(s) = \sum_{\tilde{s} \in V_{i-1}} \alpha_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s), s \in V_i$$

$$\beta_i(\tilde{s}) = \sum_{s \in V_{i+1}} \gamma_{i+1}(s, s) \beta_{i+1}(s), s \in V_i$$

Непосредственное вычисление этих величин приводит к значительными ошибкам округления, поэтому приходится ввести вспомогательные величины $\alpha'_i(s) = \frac{\alpha_i(s)}{p(y_0^{i-1})}$ и $\beta'_i(s) = \frac{\beta_i(s)}{p(y_i^{n-1}|y_{i-1})}$. Поделив $p(s_i = s', s_{i+1} = s, y_0^{n-1})$ на $\frac{p(y_0^{n-1})}{p(y_i)} = p(y_0^i|y_i)p(y_{i+1}^{n-1}|y_0^i) = p(y_0^{i-1})p(y_{i+1}^{n-1}|y_0^i)$, получим

$$p(s_i = s', s_{i+1} = s | y_0^{n-1}) p(y_i) = \frac{\alpha_i(s') \gamma_{i+1}(s', s) \beta_{i+1}(s)}{p(y_0^{i-1}) p(y_{i+1}^{n-1} | y_0^i)} = \alpha'_i(s') \gamma_{i+1}(s', s) \beta'_{i+1}(s)$$

При этом

$$L_i = \ln \frac{\sum_{(s',s) \in S_1} \alpha'_i(s') \gamma_{i+1}(s', s) \beta'_{i+1}(s)}{\sum_{(s',s) \in S_0} \alpha'_i(s') \gamma_{i+1}(s', s) \beta'_{i+1}(s)}$$

Замечание. Хотим избавиться от ошибок округления

Учитывая, что $p(y_0^{i-1}) = \sum_{x \in V_i} \alpha'_i(s)$, получим

$$\alpha'_i(s) = \frac{\alpha'_i(s)}{\sum_{s' \in V_i} \alpha'_i(s')} = \frac{\sum_{\tilde{s} \in V_{i-1}} \alpha'_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s)}{\sum_{s' \in V_i} \sum_{\tilde{s} \in V_{i-1}} \alpha'_{i-1}(\tilde{s}) \gamma_i(\tilde{s}, s')}$$

Начальные условия: $\alpha'_0(s) = \alpha_0(s) = \begin{cases} 1 & , s = 0 \\ 0 & , s \neq 0 \end{cases}$

$$p(y_i^{n-1} | y_0^{i-1}) = p(y_i^{n-1} | y_0^{i-1}) \frac{p(y_0^{i+1})}{p(y_0^{i-1})} = p(y_{i+1}^{n-1} | y_0^{i+1}) \frac{p(y_0^{i+1})}{p(y_0^{i-1})} = \frac{p(y_{i+1}^{n-1} | y_0^{i+1})}{p(y_0^{i-1})} p(y_0^{i+1}) =$$
$$= \frac{p(y_{i+1}^{n-1} | y_0^{i+1})}{p(y_0^{i-1})} \sum_{x \in V_{i+1}} \alpha_{i+1}(s) = p(y_{i+1}^{n-1} | y_0^{i+1}) \sum_{s \in V_i} \sum_{s' \in V_{i+1}} \alpha'_i(s') \gamma_{i+1}(s', s)$$

Отсюда вытекает что

$$\beta'_i(\tilde{s}) = \frac{\beta_i(s)}{p(y_i^{n-1} | y_{i-1})} = \frac{\sum_{s \in V_{i+1}} \gamma_{i+1}(\tilde{s}, s) \beta'_{i+1}(s)}{\sum_{s \in V_i} \sum_{s' \in V_{i+1}} \alpha'_i(s') \gamma_{i+1}(s', s)}$$

Начальными значениями для этой рекуррентной формулы являются

$$\beta' n(s) = \beta_n(s) = \begin{cases} 1 & , s = 0 \\ 0 & , s \neq 0 \end{cases}$$

$$\gamma_{i+1}(s', s) = p(s_{i+1} = s, y_i | s_i = s') = P\{s_{i+1} = s | s_i = s'\} p(y_i | s_i = s', s_{i+1} = s) =$$
$$= P\{c_i = \delta(s', s)\} p(y_i | c_i = \delta(s', s))$$

Вероятность $P\{c_i = \delta(s', s)\}$ представляет собой априорную вероятность того, что этот бит равен метке $\delta(s', s)$ перехода между состояниями s', s

Если рассматриваемый декодер используется как часть итеративного декодера составного кода, эта вероятность может быть найдена из апостериорных логарифмических отношений правдоподобия $L_i^{(e)}$, вычисленных другим декодером, как $P\{c_i = 1\} = \frac{\exp(L_i^{(e)})}{1+\exp(L_i^{(e)})}$, откуда следует, что

$$P\{c_i = a\} = \frac{\exp\left(\frac{L_i^{(e)}}{2}\right)}{1 + \exp(L_i^{(e)})} \exp\left((2a-1)\frac{L_i^{(e)}}{2}\right)$$

В противном случае все символы можно считать равновероятными, что эквивалентно $L_i^{(e)} = 0$

- Нахождение логарифмических отношений правдоподобия отдельных символов кодового слова $L(c_i), i = 0 \dots n-1$
- Вычисление величин $\gamma_k(s', s) = P\{c_i = \delta(s', s)\}p(y_i|c_i = \delta(s', s))$
- Вычисление $\alpha_i'(s)$ (прямая рекурсия) согласно $\alpha_i'(s) = \frac{\sum_{\tilde{s}\in V_{i-1}} \alpha_{i-1}(\tilde{s})\gamma_i(\tilde{s},s)}{\sum_{s'\in V_i}\sum_{\tilde{s}\in V_{i-1}} \alpha_{i-1}(\tilde{s})\gamma_i(\tilde{s},s')}$, $0 < i \leq n$
- Вычисление $\beta_i'(s)$ (обратная рекурсия) согласно $\beta_i'(\tilde{s}) = \frac{\sum_{s\in V_{i+1}} \gamma_{i+1}(\tilde{s},s)\beta_{i+1}'(s)}{\sum_{s\in V_i}\sum_{s'\in V_{i+1}} \alpha_i'(s')\gamma_{i+1}(s',s)}$, $0 \leq i < n$
- Вычисление апостериорных логарифмических отношений правдоподобия $L_i, 0 \leq i < n$, информационных битов согласно $L_i = \ln \frac{\sum_{(s',s)\in S_1} \alpha_i'(s')\gamma_{i+1}(s',s),\beta_{i+1}'(s)}{\sum_{(s',s)\in S_0} \alpha_i'(s')\gamma_{i+1}(s',s)\beta_{i+1}'(s)}$
- Принятие решения относительно каждого символа

Замечание. Полученная последовательность решений может не являться кодовым словом

10 Сверточные коды

10.1 Сверточные коды

Замечание. Сверточные коды преобразуют блок из k символов в блок из n символов. Преобразование зависит от предыдущих блоков.

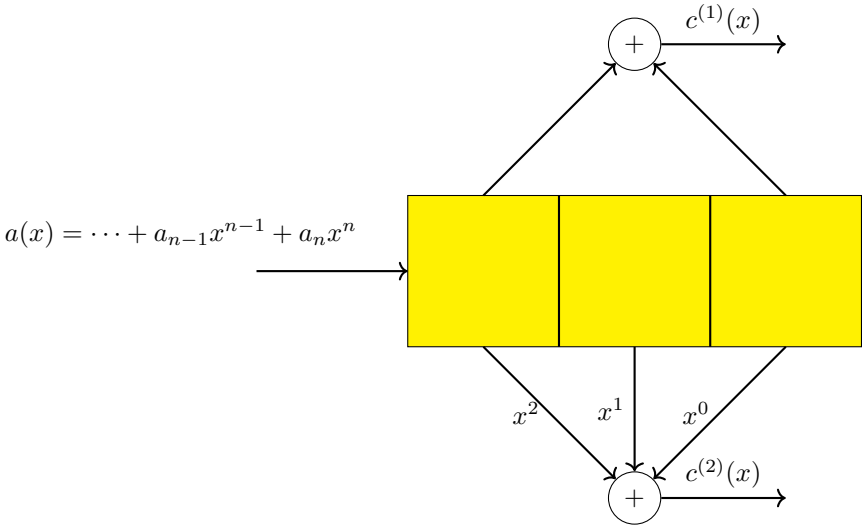
Замечание. Задача кодера – сделать передаваемые символы статистически зивисимыми

- Блоковые коды: функциональное преобразование блоков данных в кодовые слова
- Сверточные коды: автоматное отображение блоков данных в кадры колового слова

Простейший автомат – регистр сдвига. Кодер может хранить m ранее поступивших блоков из k_0 символов. На каждом шане кодер выдает n_0 закодированных символов. Скорость кода $R = \frac{k_0}{n_0}$. Объем памяти кодера – длина кодового ограничения $K = mk_0$.

Пример.

- $k_0 = 1, m = 2, K = k_0 m = 2, n_0 = 2$
- $g^{(1)}(x) = x^2 + 1, g^{(2)}(x) = x^2 + x + 1$



Замечание. $k_0 = 1$: Выходная последовательность – линейная свертка информационной последовательности и порождащих многочленов кода

$$c^{(i)}(x) = c_0^{(i)} + c_1^{(i)}x + \cdots = a(x)g^{(i)}(x) = \sum_{j\geq 0} x^j \sum_{t=0}^{m} a_{j-t}g_t^{(i)}, 1 \leq i \leq n_0$$

Теоретически кодовые слова имеют бесконечную длину. В практических системах длина кодового слова фиксирована. В конец информационной последовательности вводят несколько дополнительных битов, переводящих регистр сдвига в нулевое состояние.

Кодирование в общем случае

$$(c^{(1)}(x), \ldots, cn_0) = (a1(x), \ldots, ak_0(x))G(x)$$

$G(x)$ – $k_0 \times n_0$ порождающая матрица (многочленная) кода

Замечание. Сверточные коды являются линейными

10.2 Способы представления

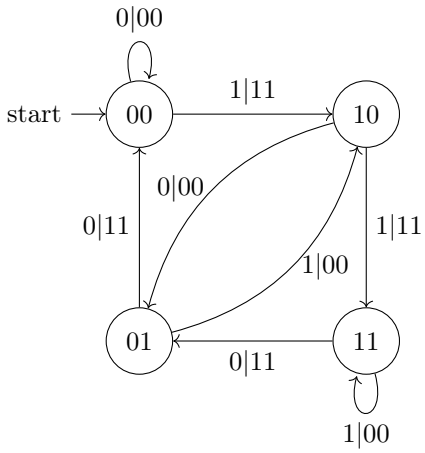
Замечание. Графическое представление сверточных кодов:

- Последовательности возможных переходов конечного автомата могут быть представлены в виде дерева. Древовидная диаграмма обладает свойством самоподобия
- Решетчатая диаграмма – боле компактный способ задания кода
- Кодовое слово – путь в решетке, начинающийся и заканчивающийся в нулевом состоянии. Фиксированная длина. Предполагается, что после обработки информационной последовательности были поданы несколько дополнительных битов, переводящих регистр сдвига в нулевое состояние

10.3 Катастрофические порождающие матрицы

Определение. Катастрофический кодер отображает информационные последовательности бесконечного веса в кодовые последовательности конечного веса

Пример. Порождающая матрица катастрофического кодера $G(x) = (x^2+1, x^2+1)$. Единичные ошибки в канале могут привести к бесконечному числу ошибок декодера. Если при передаче нулевого кодового слова возникла ошибка вида $\ldots 00011000 \ldots$, то она будет продекодирована в информационную последовательность вида $\ldots 000101010101 \ldots$, то она будет продекодирована в информационную последовательность вида $\ldots 000101010101 \ldots$



Теорема 10.1. Порождающая матрица не является катастрофической ттогда, когда НОД определителей всех $k_0 \times k_0$ подматриц $G(x)$ равен $x^s, s \geq 0$.

11 Производящая функция сверточных кодов

11.1 Производящая функция сверточных кодов

Замечание. Вероятность ошибки декодирования кода определяется числом кодовых слов различного веса. Число путей в решетке, начинающихся и заканчивающихся в нулевом состоянии

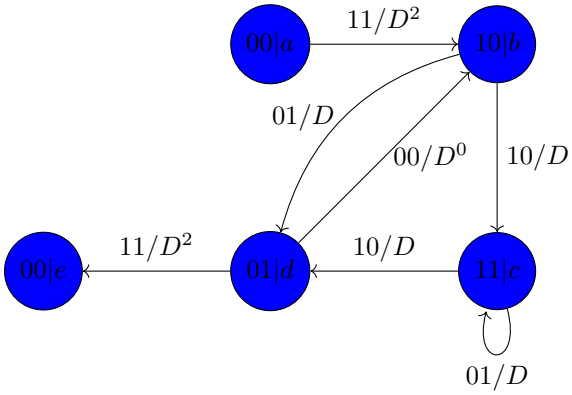
Пометим ребра графа переходов метками D^i, где i – вес кодовой последовательности. Последовательность символов веса x характеризуется одночленом D^x. Совокупность кодовых слов характеризуется многочленом, например $2D^6 + 3D^8$. Расщепим исходное состояние на два: начальное (0) и конечное (e). Пусть X_i характеризует совокупность кодовых последовательностей, приводящих кодер в состояние i. X_i – ряд, коэффициенты которого равны числу кодовых последовательностей, начинающихся в нулевом и заканчивающихся в i-ом состоянии. Производящая функция $T(D)$ равна X_e/X_a. Это степенной ряд, коэффициенты которого равны числу кодовых слов различного веса, выходящих из нулевого состояния и возвращающихся в него

Степень нунелувого члена – минимальное свободное расстояние кода. Минимальное свободное расстояние пропорционально длине кодового ограничения. При фиксированной длине блока сверточные коды хуже аналогичных блоковых

Пример.

$$\begin{cases} X_b = D^2 X_a + X_d \\ X_c = DX_b + DX_c \\ X_d = DX_c + DX_b \\ X_e = D^2 X_d \end{cases}$$

$$T(D) = \frac{D^5}{1-2D} = D^5 + 2D^6 + 4D^7 + 8D^8$$



11.2 Вероятность ошибки декодирования сверточных кодов с помощью алгоритма Витерби

Вероятность ошибки декодирования (канал с АБГШ)

$$r_{ij} = (-i)^{c_{ij}} + \eta_{ij}, \eta_{ij} \sim N(0, \sigma^2), 1 \le j \le n_0, i = 0, 1, \dots$$

Предположим, что передавалосб нулевое кодовое слово. Будем считать, что алгоритм Витерби ищет последовательность с максимальной корреляцией $C = \sum_{i \ge 0} \sum_{j=1}^{n_0} r_{ij}(-1)^{c_{ij}}$.

Оценим вероятность первого события неправильного декодирвания. Ошибка произойдет, если при слиянии нескольких путей на некотором ярусе B окажется, что $C_1 > C_0$. $_0$ – метрика ненулевого пути c_{ij}, C_1 – метрика нулевого пути

$$P\{C_1 > C_0\} = P\left\{\sum_{i=0}^{B}\sum_{j=1}^{n_0} r_{ij}((-1)^{c_{ij}} - 1) > 0)\right\} = P\left\{\sum_{i=0}^{B}\sum_{j:c_{ij}\neq 0} r_{ij} < 0\right\}$$

Объединенная верхняя граница вероятности ошибки декодирования:

- $r_{ij} \sum N(1, \sigma^2)$

Если неправильный путь имеет вес d на ярусах $0, \dots, B$, то:

$$p = \sum_{i=0}^{B}\sum_{j:c_{ij}\neq 0} r_{ij} \sim N(d, d\sigma^2), \sigma^2 = \frac{N_0}{2}$$

$$P_d = P\{p < 0\} = Q\left(\sqrt{2d\frac{E_b}{N_0}}\right) = Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right), Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt = \frac{1}{2}\mathrm{erfc}(x/\sqrt{2})$$

Вероятность ошибки – вероятность того, что будет выбран какой-либо неправильный путь

$$P_C = P\{(C_1 > C_0) \vee (C_2 > C_0) \vee \dots\} \le \sum_i P\{C_i > C_0\} = \sum_{d>0} t_d P_d = \sum_{d=d?}^{\infty} t_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right)$$

Производящая функция $T(D) = \sum_{d \ge 0} t_d D^d$

Вероятность ошибки на бит в случае выборп ошибочного пути C_i с кодовой и информационной последовательности отичающихся от истснных в d и w позициях, соответсвенно, равна $\frac{w}{k_0}P\{C_i > C_0\} = \frac{w}{k_0}P_d$. Расширенная производящая функция $T(N, D) = \sum_{w,d} t_{wd} N^w D^d$.

$$t(D) = \left.\frac{\partial T(N,d)}{\partial N}\right|_{N=1} = \sum_d D^d \underbrace{\sum_w t_{wd} w}_{b_d}$$

Общая вероятность ошибки декодирования на бит

$$P_b \le \frac{1}{k_0}\sum_{d=d_{f?}}^{\infty} b_d Q\left(\sqrt{2dR\frac{E_b}{N_0}}\right)$$

12 Комибинирование кодов

12.1 Конструкция Плоткина

Теорема 12.1. Пусть даны (n, k, d) коды $C_i, i = 1, 2$, $C = \{(c_1, c_1 + c_2) | c_i \in C_i, i = 1, 2\}$ – $(2n, k_1 + k_2, \min(2d_1, d_2))$ код

12.2 Коды Рида-Маллера

Замечание. Рекурсивное применение конструкции Плоткина

- $RM(r, m)$ – код Рида-Маллера порядка r длины 2^m
- $RM(0, m) = (2^m, 1, 2^m)$
- $RM(m, m) = (2^m, 2^m, 1)$
- $RM(r + 1, m + 1)$ применение конструкции Плоткина к $C_1 = RM(r + 1, m), C_2 = RM(r, m)$

Определение. Код Рида-Маллера $RM(r, m)$ длины 2^m порядка r – полярный код с $\mathcal{F} = \{i | 0 \le i < 2^m, wt(i) < m - r\}$

12.3 Прямое произведение кодов

Определение. Пусть даны (n_1, k_1, d_1) (кодирование по строчкам) и (n_2, k_2, d_2) (кодирование по столбцам) коды с порождающими матрицами G', G''. Кодовое слово образуется путем выписывания полученной матрицы по столбцам. $(n_1 n_2, k_1 k_2, d_1 d_2)$ код с порождающей матрицей

$$G' \otimes G'' = \begin{pmatrix} G'_{11}G'' & G'_{12}G'' & \dots & G'_{1n_1}G'' \\ G'_{21}G'' & G'_{22}G'' & \dots & G'_{2n_1}G'' \\ \vdots & \vdots & \ddots & \vdots \\ G'_{k_11}G'' & G'_{k_12}G'' & \dots & G'_{k_1n_1}G'' \end{pmatrix}$$

$$\mathrm{R} = \mathrm{k}_1 k_2 \overline{{}_{n_1 n_2 < \frac{k_1}{n_1} \cdot \frac{k_2}{n_2}}}$$

12.4 Обобщенные каскадные коды

Определение. Внешние (N_i, K_i, D_i) коды $\mathcal{A}_i$ над $GF(q^{m_i}), 1 \le i \le s$.

Замечание. Вложенные внутренние (n_i, k_i, d_i) коды $\mathcal{B}_i : B_1 \supset \mathcal{B}_2 \supset \dots \supset \mathcal{B}_s$ над $GF(q)$

- $k_i - k_{i+1} = m_i$
- Код $\mathcal{B}_i$ порождается последними k_i строками $k_1 \times n$ матрицы B

Кодирование:

- Закодируем данные внешними кодами и запишем полученные кодовые слова в $s \times N$ матрицу X
- Заменим элементы i-ой строки X на их векторное представление (столбец длиной m_i). Пусть Y – полученная $k_1 \times N$ матрица
- Умножим каждый столбец Y на B (т.е. закодируем в коде $\mathcal{B}_1$)
- Полученная $n \times N$ матрица может рассматриваться как кодовое слово

Линейный $(Nn, \sum_{i=1}^{s} K_i m_i, \ge \min_{1 \le i \le s} d_i D_i)$ код над $GF(q)$. Некоторые ОКК (полярные коды) достигают предела Шеннона.

Доделать Картинка

- Запишем принятые символы в виде $n \times N$ матрицы
- `for i=1, ... , s`
  - Продекодируем столбцы в коду $\mathcal{B}_i$
  - Продекодируем i-ую строку в коде $\mathcal{A}_i$. Пусть $(c_i, \dots, c_N)$ – найденное кодовое слово
  - Вычтем из j-ого столбца $c_j B_i^T$

13 Модификации

13.1 Укорочение, выкалывание и расширение кодов

Определение. Укороченный код полсучается путем выбора кодовых слов исходного кода, содержащих нули на заданных позициях, с последующим удалением этих нулей

Пусть дан (n, k, d) код с порождающей матрицей $G = (I|A)$. Удалим из порождающей матрицы m столбцев удиничной подматрицы и соответствующие m строк.

Определение. Выколотый код: удалим из всех кодовых слов заданные символы (как правило, проверочных).

Пусть дана проверочная матрица (n, k, d) кода в форме $H = (A| - I)$. Удалим из H m столбцов единичной подматрицы и соответствующие им m строк. Если проявятся линейно зависимые строки, удалим их $\implies$ $(n - m, \le k, \ge d - m)$ код

Определение. Наиболее распространенный способ – добавление проверки на четность. $(n, k, d) \Rightarrow (n+1, k, d')$.

Если минимальное расстояние d исходного кода нечетно, то минимальное расстояние расширенного кода $d' = d + 1$.

Пример. $(7, 4, 3)$ код Хемминга

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

$(8, 4, 4)$ расширенный код Хемминга

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

13.2 Турбо-коды

Определение. Одновременное кодирование информационных битов нескольких сверточными кодами позволяет исключить многие конфигурации ошибок, которые могли бы возникнуть при использовании независимых сверточных кодов. Должны использоваться рекурсивные систематические сверточные коды. Многие информационные последовательности маого веса превращаются в кодовые слова большого веса. Длина кодового ограничения должна быть небольшой. Кодам с большим кодовым ограничением присущи длинные пакеты ошибок декодирования. Турбо-код является линейным блоковым кодом. Для повышения скорости кода используется выкалывание

Одновременное кодирование информационных битов нескольких сверточными кодами позволяет исключить многие конфигурации ошибок, которые могли бы возникнуть при использовании независимых сверточных кодов. Должны использоваться рекурсивные систематические сверточные коды. Многие информационные последовательности маого веса превращаются в кодовые слова большого веса. Длина кодового ограничения должна быть небольшой. Кодам с большим кодовым ограничением присущи длинные пакеты ошибок декодирования. Турбо-код является линейным блоковым кодом. Для повышения скорости кода используется выкалывание

Минимальное расстояние Доделать Картинка

Доделать Картинка

Декодеры сверточных кодов входящих в турбо-код, обмениваются информацией, полученной в результате декодирования. Как правило, достаточно $5 - 10$ итераций. Апостериорные логарифмические отношение правдоподобия информационных символов, вычисленные одним декодером, являются априорными ЛОПП для другого декодера. Аппроксимация декодера максимального правдоподобия. Этот подход применим и для декодирования прямого произведения кодов

14 Полярные коды

14.1 Параметр Бхаттачарьи

Определение. $W(y|c)$ – вероятность наблюдения на выходе канала $y \in \mathcal{Y}$ при условии подачи на его вход $c \in \mathcal{X}$

Определение. Рассмотрим приемник по максимуму прадоподобия $\tilde{c} = \text{argmax}_{c \in 0,1} W(y|c)$. Передаваемые символы рановероятны. Вероятность ошибки

$$P_c = P\{c=0\}P\{err|e=0\} + P\{c=1\}P\{err|c=1\} =$$
$$= \frac{1}{2}\sum_{y:W(y|0)<W(y|1)} W(y|0) + \frac{1}{2}\sum_{y:W(y|1)<W(y|0)} W(y|1) =$$
$$= \frac{1}{2}\sum_{y:\frac{W(y|1)}{W(y|0)}>1} W(y|0) + \frac{1}{2}\sum_{y:\frac{W(y|0)}{W(y|1)}>1} W(y|1) =$$
$$= \frac{1}{2}\sum_{y\in\mathcal{Y}}\sum_{c\in\{0,1\}}\left(W(y|c)\chi\left(\frac{W(y|1-c)}{W(y|c)}\right)\right)$$

Индикаторная фунукция $\chi(z) = \begin{cases} 1 & ,z \geq 1 \\ 0 & ,z < 1 \end{cases}$

$$P_c \leq \frac{1}{2}\sum_{y\in\mathcal{Y}}\sum_{c\in\{0,1\}}\left(W(y|c)\chi\left(\frac{W(y|1-c)}{W(y|c)}\right)\right) = \sum_{y\in\mathcal{Y}}\sqrt{W(y|0)W(y|1)} = Z(W)$$

Определение. Параметры Бхаттачарьи битовых подканалов $Z_{m,i} = Z(W_m^{(i)})$

$$Z_{m,2i+1} \leq Zm, 2i \leq 2Z_{m-1,i} - Z_{m-1,i}^2$$

$$Z_{m,2i+1} = Z_{m-1,i}^2$$

Строгое равенство в случае двоичного стирающего канала

14.2 Поляризация канала

Замечание. Пропускные способности битовых подканалов $I_{m,i} = I(W_m^{(i)})$

$$I_{m,2i} + I_{m,2i+1} = 2I_{m-1,i}$$

$$I_{m,2i} \leq I_{m,2i+1}$$

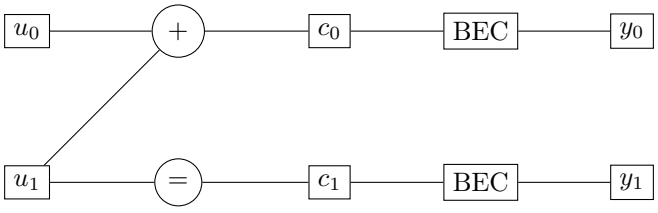
$$\sqrt{1 - Z(W)^2} \geq I(W) \geq \log\frac{2}{1+Z(W)}$$

Для любого $\delta \in (0,1)$ при $m \to \infty$ доля подканалов с $I(W_m^{(i)}) \in (1-\delta, 1]$ стермиться к $I(W_0^{(0)}) - I(W)$, а доля подканалов с $I(W_m^{(i)}) \in [0,\delta)$ стремится к $1 - I(W)$

Замечание. Поляризация каналов: Доделать Картинка

- Доля недополяризованных подканалов убывает с увеличением m
- Число недополяризованных подканалов растет с увеличением m

Определение. Рассмотрим линейное преобразование, задаваемое



$$\begin{pmatrix} c_0 & c_1 \end{pmatrix} = \begin{pmatrix} u_0 & u_1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Двоичный стирающий канал: $y = \begin{cases} c_i & ,\text{с вероятностью } 1-p \\ e & ,\text{с вероятностью } p \end{cases}$

- u_0 не может быть восстановлен из y_0, y_1 с вероятностью $1-(1-p)^2 = 2p - p^2 \geq p$
- u_1 не может быть восстановлен из u_0, y_0, y_1 с вероятностью $p^2 \leq p$

14.3 TODO Полярные коды

Замечание. Передавать предопределенные значения (например, 0) по плохим подканалам. Кодирование $c_0^{n-1} = u_0^{n-1} A_m, u_i = 0, i \in \mathcal{F}$, где $\mathcal{F}$ – множество номеров плохих подканалов (замороженных символов). Линейный блоковый код $(2^m, 2^m + |\mathcal{F}|)$

14.4 Сложность кодирования

$$u_0^{n-1} A_m = \begin{pmatrix} u_0^{n/2-1} & u_{n/2}^{n-1} \end{pmatrix} \begin{pmatrix} A_{m-1} & 0 \\ A_{m-1} & A_{m-1} \end{pmatrix} = \begin{pmatrix} (u_0^{n/2-1} + u_{n/2}^{n-1}) A_{m-1} & u_{n/2}^{n-1} A_{m-1} \end{pmatrix}$$

Сложность $T(n) = 2T(n/2) + n/2 = \frac{1}{2} n \log_2 n$

15 Алгоритм последовательного исключения и декодер Тала-Варди

15.1 Алгоритм последовательного исключения и декодер Тала-Варди

Program 2 Алгоритм последовательного исключения

1: **for** $i = 0, 1, \ldots, 2^m - 1$ **do**
2: $\quad \hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ \operatorname{argmax}_{u_i} W_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1} | u_i) & , i \notin \mathcal{F} \end{cases}$
3: **end for**

Program 3 Алгоритм последовательного исключения с ЛОПП

1: **for** $i = 0, 1, \ldots, 2^m$ **do**
2: $\quad \hat{u}_i = \begin{cases} 0 & , i \in \mathcal{F} \\ 0 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) > 0, i \notin \mathcal{F} \\ 1 & , L_m^{(i)}(y_0^{n-1}, \hat{u}_0^{i-1}) \leq 0, i \notin \mathcal{F} \end{cases}$
3: **end for**

16 Построение полярных кодов

16.1 TODO Построение полярных кодов

Замораживанию подлежат $2^m - k$ наименее надежных символов (например, с наибольшим $Z_{m,i}$. Двоичный стирающий канал

$$Z_{m,2i} = 2Z_{m-1,i} - Z_{m-1,i}^2$$

$$Z_{m,2i+1} = Z_{m-1,i}^2$$

Сложность вычисления $Z_{m,i} = O(n)$. В общем случае выходной алфавит канала $W_m^{(i)}(y_0^{n-1}, u_0^{i-1} | u_i)$ имеет мощность $|\mathcal{Y}|^n 2^i$. Построение функции переходных вероятностей $W_m^{(i)}$ вычислительно нереализуемо уже при небольших m. Можно аппроксимировать канал $W_m^{(i)}$ каналом с выходным алфавитом фиксированной мощности μ, который был бы чуть лучше или чуть хуже, чем истинный $W_m^{(i)}$. $Z_{m,i}$ могут быть вычисленны со сложностью $O(n\mu^2 \log \mu)$.

16.2 Полярные коды с CRC, полярные подкоды

Определение. Cyclic redundancy check – циклический код, обнаруживающий ошибки

Замечание. Систематическое кодирование циклического кода длины n с порождающим многочленом $g(x)$

$$c(x) = a(x)x^{n-k} + b(x), b(x) \equiv a(x)x^{n-k} \mod g(x)$$

Добавим к данным проверочные символы ($b(x)$ перед их кодированем полярным кодом. Удалим из списка, формируемого декодером Тала-Варди кодовые слова с направильным значением контрольной суммы.

16.2.1 Полярные коды в узком смысле

- Выберем линейный блоковый код (протокод) с достаточно большим минимальным расстоянием
- Удалим из него кодовые слова, препятствующие эффективному декодированию методом последовательного исключения

Определение. Рассмотрим канал $W(y|c)$ и $(n = 2^m, k', d)$ код $\mathcal{C}'$ над $GF(2)$, называемый протокодом. Пусть $\mathcal{F}'$ – множество номеров замороженных символов $\mathcal{C}'$. $(n, k, \geq d)$ полярным подкодом $\mathcal{C}$ в узком смысле кода $\mathcal{C}'$ называется множество векторов $c_0^{n-1} = u_0^{n-1} A_m$, где u_0^{n-1} одновременно удовлетворяет ограничениям замораживания кода $\mathcal{C}'$, а также дополнительным ограничениям $u_s = 0$ для $k' - k$ номеров $s \notin \mathcal{F}'$ с наибольшими вероятностями ошибки $P_{m,s}$.

Замечание. Расширенные примитивные коды БЧХ в узком смысле – хорошие протокоды

Матрица ограничений и матрица прекодирования:

$$c_0^{n-1} = u_0^{n-1} A_m, \quad u_0^{n-1} V^T = 0$$

$$u_0^{n-1} = xW, \quad WV^T = 0$$

- V – матрица ограничений (аналог проверочной матрицы)
- W – матрица прекодирования (аналог порождающей матрицы)

16.2.2 Полярные коды в широком смысле

- Выберем полярный код $(n = 2^m, n - r)$, эффективно декодируемый методом ПИ
- Удалим их него кодовые слова, ответственные за высокую вероятность ошибки декодирования МП, построив его подкод размерности $k < n - r$.

Определение. Полярным кодом в широком смысле называется множество векторо

$$c = x\mathbb{W}A_m, x \in FG(2)^k$$

где матрица $\mathbb{W}$ имеет нулевые стобцы в позициях, соответствующих r нименее надежным подканалам $W_m^{(j)}$

17 Циклические коды

17.1 Циклические коды

Определение. Линейный блоковый код $\mathcal{C}$ длины n над полем $\mathbb{F}$ называется циклическим, если любой циклический сдвиг его кодового слова также является кодовым словом, т.е. $(c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C} \implies (c_{n-1}, c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}$

Замечание. Многочленное представление вектора $(c_0, c_1, \ldots, c_{n-1}) : c(x) = \sum_{i=0}^{n-1} c_i x^i$. Циклический свдиг вектора на одну позицию эквивалентен

$$xc(x) = xc_0 + x^2 c_1 + \cdots + c_{n-2}x^{n-1} + c_{n-1}x^n \equiv c_{n-1} + xc_0 + x^2 c_1 + \cdots + c_{n-2}x^{n-1} \mod x^n - 1$$

В дальнейшем вектор $(c_0, c_1, \ldots, c_{n-1})$ и соответствующией многочлен $c(x)$ будут считаться равнозначными

Теорема 17.1. Подмножество $\mathcal{C} \subset \mathbb{F}[x] \setminus (x^n - 1)$ образует циклический код ттогда, когда:

1. $\mathcal{C}$ образует группу по сложению
2. Если $c(x) \in \mathcal{C}$ и $a(x) \in \mathbb{F}[x] \setminus (x^n - 1)$, то $[a(x)c(x) \mod (x^n - 1)] \in \mathcal{C}$

17.2 Порождающий и проверочный многочлены

Замечание.

- Порождающий многочлен циклическоо кода – ненулевой кодовый многочлен $g(x) \in \mathcal{C}$ нименьшей степени с коэффициентов при старшем члене 1
- Все кодовые слова $c(x)$ в ЦК делятся на $g(x)$

  Предположим противное $\implies c(x) = a(x)g(x) + r(x), r(x) \in \mathcal{C}$. $\deg r(x) < \deg g(x)$, что противоречит предположению о минимальности степени $g(x)$
- Порождающий многочлен циклического кода единственен
- ЦК длины n с ПМ $g(x)$ сущесвует ттогда, когда $g(x) | (x^n - 1)$
  - Существует код $\mathcal{C}$ с ПМ $g(x) \implies$
    - $x^n - 1 = a(x)g(x) + r(x), \deg r(x) < \deg g(x)$
    - $b(x) \equiv a(x)g(x) \mod (x^n - 1), b(x) \in \mathcal{C}$
    - $r(x) = (x^n - 1 - a(x)g(x)) \equiv -a(x)g(x) \mod (x^n - 1), r(x) \in \mathcal{C} \implies r(x) = 0$
  - $\Leftarrow$: в качестве порождающего многочлена можно выбрать любой делитель $x^n - 1$
- $(x^n - 1) = h(x)g(x), h(x)$ – проверочный многочлен кода
- Для любого $c(x) \in \mathcal{C} : c(x)h(x) = a(x)g(x)h(x) \equiv 0 \mod x^n - 1$
- Размерность циклического кода равна $kd \deg h(x)$

17.3 Кодирование

Замечание. Несистематическое кодирование $c(x) = a(x)g(x)$

$$(c_0, \dots, c_{n-1}) = (a_0, \dots, a_{k-1}) \begin{pmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & \dots & 0 \\ 0 & g_0 & g_1 & g_2 & \dots & g_{n-k} & \dots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \dots & g_{n-k} \end{pmatrix}$$

Замечание. Систематическое кодирование (информационные символы $a_0, \dots, a_{k-1}$ в $c_{n-k}, \dots, c_{n-1}$)

$$c(x) = x^{n-k}a(x) - r(x)$$

$$r(x) \equiv x^{n-k}a(x) \mod g(x), \deg r(x) < \deg g(x)$$

Каждому методу кодирования соответствует своя порождающая матрица. Все порождающие матрицы выражаются друг через друга как $G' = QG$, где Q – обратимая матрица. Используемый метод кодирования не влияет на корректирующую способность кода

18 Поля

18.1 Идеалы, факторкольца и поля

Определение. Группа $\mathcal{G}$ – алгебра $(G, \cdot)$.

- Операция $\cdot$ ассоциативна, т.е. $\forall a, b, c \in G : (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Существует нейтральный элемент $\mathbb{1} \in G : \forall x \in G : \mathbb{1} \cdot x = x \cdot \mathbb{1} = x$
- Существует обратный элемент $\forall x \in G \exists y : x \cdot y = y \cdot x = \mathbb{1}$

Определение. Подмножество I кольца R называется **правым (или левым) идеалом**, если

- $(I, \{+\})$ является подгруппой $(R, \{+\})$

- $\forall r \in R, \forall x \in I : xr \in I$ (или $rx \in I$)

Определение. Если $A \subset R$, то идеалом, порождаемым A, называется нименьший идеал, содержащий A:

$$\langle A \rangle = \sum_i r_i a_i, a_i \in A, r_i \in R$$

Определение. Пусть дано кольцо R и идеал $I \subset R$. Бинарное отношение $\sim\equiv \{(a,b) \in R^2 | b - a \in I\}$ является отношением конгруэнтности. Разбиение на классы эквивалентности $[a] = a + I = a \mod I = \{a + r | r \in I\}$. Множество классов эквивалентности по отношению $\sim$ называется **факторкольцом** или кольцом вычетов R по модулю I и обозначается $R \setminus I$.

Замечание. Факторкольцо является кольцом, если на нем определить операции следующим образом:

- $(a + I) + (b + I) = \underbrace{(a + b)}_{\in R} + I$
- $-(a + I) = (-a) + I$
- $(a + I) \cdot (b + I) = ab + I$
- Нулевой элемент $\mathbb{0} + I = I$, единичный элемент $\mathbb{1} + I$

Определение. Полем называется алгебра $(\mathbb{F}, \{+, -, \cdot, /\})$, удовлетворяющая следующим аксиомам

- $a + (b + c) = (a + b) + c$
- $a + b = b + a$
- ($\exists$ 0$\in$ F: a + 0= a\)
- $\forall a \in F : \exists a' = -a \in \mathbb{F} : a + a' = \mathbb{0}$
- $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- $a \cdot b = b \cdot a$
- $a \cdot (b + c) + a \cdot b + a \cdot c, (b + c) \cdot a = b \cdot a + c \cdot a$
- $\exists \mathbb{1} \in \mathbb{F} \setminus \{\mathbb{0}\} : \forall a : a \cdot \mathbb{1} = \mathbb{1} \cdot a = a$
- $\forall a \neq \mathbb{0} \exists a^{-1} = \mathbb{1}/a \in \mathbb{F} : a \cdot a^{-1} = a^{-1} \cdot a = \mathbb{1}$

19 Конечные поля

19.1 Конечные поля

Замечание.

- Поля, содержащие конечное число q элементов – конечные поля или поля Галуа \(GF(q))
- Бесконечные поля называются полями характеристики нуль
- Поле является областью целостности, т.к. если допустить, что $\exists a, b \neq 0 : ab = \mathbb{0}$, то $\mathbb{0} = (((ab)b^{-1})a^{-1}) = \mathbb{1}) = 1$\)
- Простое поле $GF(p)$:
  - p – простое
  - арифметика по модулю p

19.2 Характеристика поля

Определение. Если поле кончено, то не могут быть различными все элементы $\mathbb{1}, \mathbb{1}+\mathbb{1}, \mathbb{1}+\mathbb{1}+\mathbb{1}, \ldots$. Следовательно, существует наименьшее число $p : \underbrace{\mathbb{1} + \mathbb{1} + \cdots + \mathbb{1}}_{p \text{ раз}} = \mathbb{0}$. p – **характеристика поля**

19.3 Основные свойства

Замечание.

- Для всякого ненулевого $\beta \in GF(q)$ выполняется $\beta^{q-1} = 1$
- Все элементы поля $GF(q)$ удовлетворяют уравнению $x^q - x = 0$
- Порядок любого ненулевого $\beta \in GF(q)$ делит $q - 1$
- В поле характеристики $p > 1$ справедливо

$$(x + y)^p = x^p + y^p$$

$$(x + y)^p = \sum_{i=0}^{p} C_p^i x^i y^{p-1}; C_p^i = \frac{p!}{i!(p-i)!} \equiv 0 \mod p, 0 < i < p$$

19.4 Число элементов в поле Галуа и его примитивный элемент

Теорема 19.1. Пусть $\mathbb{F}$ – поле из q элементов. Тогда $q = p^m$, где p – простое, а $m \in \mathbb{N}$

20 Минимальные многочлены

20.1 Минимальные многочлены

Определение. Минимальными многочленом элемента $\beta \in GF(p^m)$ над $GF(p)$ называется нормированный многочлен $M_\beta(x) \in GF(p)[x]$ наименьшей сетепени, т.ч. $M_\beta(\beta) = 0$

Теорема 20.1. $M_\beta(x)$ неприводим над $GF(p)$

Теорема 20.2. Если $f(x) \in GF(p)[x]$ и $f(\beta) = 0$, то $M_\beta(x) | f(x)$ ($f(x)$ делится на этот минимальный многочлен)

Теорема 20.3. $M_\beta(x) | (x^{p^m} - x)$ для $\beta \in GF(p^m)$

Теорема 20.4.

Теорема 20.5. Если α – примитивный элемент $GF(p^m)$, то степень его минимального многочлена равна m

Теорема 20.6. Все конечные поля $GF(p^m)$ изоморфны

Теорема 20.7. $\forall \beta \in GF(p^m) : M_\beta(x) = M_{\beta^p}(x)$

Теорема 20.8. $M_\beta(x) = \prod_{i=0}^{m_\beta - 1}(x - \beta^{p^i})$, где m_β – наименьшее положительное число, т.ч. $\beta^{p^{m_\beta - 1}} = \beta$

21 Коды БЧХ

21.1 Проверочная матрица циклического кода над расширенным полем

Замечание. Пусть порождающий многочлен циклического кода $\mathcal{C}$ над $GF(q)$ имеет корни $\beta_1, \ldots, \beta_r \in GF(q^m) \implies \forall c(x) \in \mathcal{C} : c(\beta_i) = 0, 1 \le i \le r \implies \sum_{j=0}^{n-1} c_j \beta_i^j = 0 \implies Gc^T = 0$

$$\begin{pmatrix} \beta_1^0 & \beta_1^1 & \ldots & \beta_1^{n-1} \\ \beta_2^0 & \beta_2^1 & \ldots & \beta_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_r^0 & \beta_r^1 & \ldots u \beta_r^{n-1} & \end{pmatrix}$$

Проверочная матрица H' над $GF(q)$: заменить $\beta_i^j \in GF(q^m)$ на вектора-столбцы длины m из $GF(q)$, соответствующие их разложению по некоторому базису $GF(q^m)$

21.2 Граница БЧХ

Теорема 21.1. Если порождающий многочлен циклического кода длины n над $GF(q)$ имеет корни $\alpha^b, \dots, \alpha^{b+\delta-1}$, где $\alpha \in GF(q^m)$ – примитивный корень степени n из 1, то минимальное расстояние этого кода $d \geq \delta$

21.3 Коды БЧХ

Определение. Кодом БЧХ над $GF(q)$ длины n с конструктивным расстоянием δ называтеся циклический код наибольшей возможной размерности, порождающий многочлен которого имеет корни $\alpha^b, \dots, \alpha^{b+\delta-2}$, где $\alpha \in GF(q^m)$ – примитивный корень степени n из 1

Замечание. В силу теоремы Лагранжа $n|(q^m - 1)$. Если невозможно подобрать такое m соответствующего кода БЧХ не существует

21.4 Свойства

Замечание. Размерность кода БЧХ $k \geq n - m(\delta - 1)$

- Проверочная матрица над $GF(q^m)$ содержит $\delta - 1$ строк
- Проверочная матрица над $GF(q)$ содержит $m(\delta - 1)$ строк. Некоторые их них могут быть линейно зависимы

Замечание. Двоичные коды БЧХ в узком смысле $(b = 1) : k \geq n - m\lfloor (d-1)/2 \rfloor$

- $M_\beta(x) = M_{\beta^2}(x)$
- $g(x) = \text{LCM}(M_{\alpha^1}(x), M_{\alpha^3}(x), \dots, M_{\alpha^{\delta-2}}(x))$
- В проверочную матрицу над $GF(2^m)$ достаточно включить $\left\lfloor \frac{d-2}{2} \right\rfloor$ строк, соответствующих α^{2i+1}

Замечание. Рассмотрии исправление ошибок в векторе $y = c + e$.

- $y(x) = a(x)g(x) + e(x)$
- Синдром: $S_i = y(\alpha^{b+i}) = a(\alpha^{b+i})g(\alpha^{b+i}) + e(\alpha^{b+i}) = e(\alpha^{b+i}), 0 \leq i < \delta - 1$
- Пусть ошибки произошли в позициях $j_1, \dots, j_t, t \leq \lfloor (\delta - 1)/2 \rfloor$

$$S_i = \sum_{r=0}^{n-1} e_r \alpha^{(b+i)r} = \sum_{l=1}^{t} c_{j_l} \alpha^{(b+i)j_l}$$

- Значение ошибок $E_l = e_{j_l}$
- Локаторы ошибок $X_l = \alpha^{j_l}$

$$S_i = \sum_{l=1}^{t} E_l X_l^{b+i}, 0 \leq i < \delta - 1$$

22 Коды Рида-Соломона

22.1 Коды Рида-Соломона и обобщенные коды Рида-Соломона

Определение. Код Рида-Соломона – код БЧХ длины $q-1$ над $GF(q)$. Минимальный многочлен $\beta \in GF(q)$ над $GF(q)$: $M_\beta(x) = x - \beta$. Порождающий многочлен кода Рида-Соломона $g(x) = \prod_{i=0}^{\delta-2}(x - \alpha^{b+i})$. Размерность кода $k = n - \delta + 1$. Минимальное расстояние $d \geq \delta$. Граница Синглтона: $d \leq n - k + 1 = \delta \implies d = n - k + 1$. Код с максимальным достижимым расстоянием

Определение. $n, k, n-k+1$ кодом Рида-Соломона называется множество векторов $c = (c_0, \dots, c_{n-1})$, где $c_i = f(a_i), \deg f(x) < k, f(x) \in GF(q)[x], a_i \in GF(q)$ – различные значения (локаторы)

Определение. Обобщенным $(n, k, d = n - k + 1)$ кодом Рида-Соломона $GRS(n, k, a, u)$ называется множество векторов $(c_0u_0, \dots, c_{n-1}u_{n-1})$, где $(c_0, \dots, c_{n-1})$ – кодовое слово $(n, k, n - k + 1)$ кода Рида-Соломона (т.е. $c_i = f(a_i), \deg f(x) < k, a_i$ – различные), и $u_0, \dots, u_{n-1}$ – ненулевые константы

23 Алгоритм Питерсона-Горенстайна-Цирлера

23.1 Алгоритм Питерсона-Горенстайна-Цирлера декодирования кодов БЧХ

$$\Lambda_1 S_{j+t-1} + \dots + \Lambda_t S_j = -S_{j+t}$$

$$\underbrace{\begin{pmatrix} S_0 & S_1 & \dots & S_{t-1} \\ S_1 & S_2 & \dots & S_t \\ \vdots & \vdots & \ddots & \vdots \\ S_{t-1} & S_t & \dots & S_{2t-2} \end{pmatrix}}_{\mathbb{S}_t} \begin{pmatrix} \Lambda_t \\ \Lambda_{t-1} \\ \vdots \\ \Lambda_1 \end{pmatrix} = \begin{pmatrix} -S_t \\ -S_{t+1} \\ \vdots \\ -S_{2t-1} \end{pmatrix}$$

- Вычисление синдрома $S_i = y(\alpha^{b+i}), 0 \le i < \delta - 1$. Сложность при использовании схемы Горнера $O((\delta-1)n)$
- Будем уменьшать предполагаемое число ошибок $t \le \tau = \lfloor(\delta-1)/2)\rfloor$, пока матрица $\mathbb{S}_t$ не станет обратимой. Проверка обратимости матрицы требует $O(t^3)$ операций
- Решение СЛАУ задает коэффициенты $\Lambda_i, 1 \le i \le t$, многочлена локаторов ошибок $\Lambda(x) = 1 + \sum_{i=1}^{t} \Lambda_i x^i$
- Сложность непосредственного подбора t и решения СЛАУ $O(\tau^4)$
- Локаторы ошибок $X_i = \alpha^{j_i} : \Lambda(X_i^{-1}) = 0, 1 \le i \le t$. Процедура Ченя поиска корней: подставим в $\Lambda(x)$ все элементы $\alpha^i, 0 \le i < n$. Сложность $O(nt)$
- Значения ошибок $E_l : S_i \sum_{l=1}^{t} E_l X_l^i, 0 \le i < t$. Сложность непосредственного решения $O(t^3)$

$$S_i = \sum_{l_1}^{t} E_l X_l^{b+i}, 0 \le i < \delta - 1 \quad S(x) = \sum_{i=0}^{\delta-2} S_i x^i = \sum_{l=1}^{t} E_l X_l^b \sum_{i=0}^{\delta-2} (X_l x)^i$$

$$\text{1 - } (\mathrm{X}_l x)^{\delta-1} d(1 - X_l x) \left(\sum_{i=0}^{\delta-2} (X_l x)^i\right) = 1 \mod x^{\delta-1}$$
$$\sum_{i=0}^{\delta-2} (X_l x)^i = \frac{1}{1 - X_l x} \mod x^{\delta-1}$$
$$\mathrm{S(x)} = \sum_{l=1}^{t} \frac{E_l X_l^b}{1 - X_l x} \mod x^{\delta-1}$$

Многочлен значений ошибок $\Gamma(x) = \sum_{l=1}^{t} E_l X_l^b \prod_{j \ne l} (1 - X_j x) \equiv \Lambda(x) \sum_{l=1}^{t} \frac{E_l X_l^b}{1 - X_l x} \mod x^{b-1}$.

$$\Gamma(x) \equiv \Lambda(x) S(x) \mod x^{\delta-1}, \deg \Lambda(x) \le \lfloor(\delta-1)/2\rfloor, \deg \Gamma(x) < \lfloor(\delta-1)/2\rfloor$$

Теорема 23.1 (Алгоритм Форни быстрого поиска значений ошибок)**.** $E_i = \frac{X_i^{-b} \Gamma(X_i^{-1})}{\prod_{j \ne i}(1 - X_j X_i^{-1})}, 0 \le i < t$

Доказательство.

$$\frac{X_i^{-b}(X_i^{-1})}{\prod_{j \ne i}(1 - X_j X_i^{-1})} = \frac{X_i^{-b} \sum_{l=1}^{t} E_l X_l^b \prod_{j \ne i}(1 - X_j X_i^{-1})}{\prod_{j \ne i}(1 - X_j X_i^{-1})} = \frac{E_i \prod_{j \ne i}(1 - X_j X_i^{-1})}{\prod_{j \ne i}(1 - X_j X_i^{-1})} = E_i$$

□

Сложность $O(t^2)$

24 Декодирование БЧХ

24.1 Ключевое уравнение декодирования кодов БЧХ

- Вычисление синдрома
  - Схема Горнера: $S_i = y(\alpha^{b+i}) = y_0 + \alpha^{b+i}(y_1 + \alpha^{b+i}(y_3 + \dots))), 0 \le i < \delta$. Сложность $O(n\delta)$ операций

  - Метод Герцеля: $r_i(x) \equiv y(x) \mod M_{\alpha^{b+i}}(x)$; $S_i = r_i(\alpha_{b+i}), \alpha \in GF(p^m)$. $M_{\alpha^{b+i}} \in GF(p)[x]$ – минимальный многочлен α^{b+i}. Деление на него тербует только сложений. Минимальные многочлены многих α^{b+i} совпадают

- Решение ключевого уравнения $\Gamma(x) \equiv S(x)\Lambda(x) \mod x^{\delta-1}$. Расширенный алгоритм Евклида: $O(\delta^2)$ операций
- Поиск корней X_i^{-1} многочлена локаторов ошибок $\Lambda(x)$. Процедура Ченя (перебор $\alpha^i, 0 \le i < n$ и проверка $\Lambda(\alpha^i) = 0$) со сложностью $O(n\delta/2)$
- Поиск значений ошибок. Метод Форни со сложностью $O(\delta^2)$

24.2 Алгоритм Форни

Теорема 24.1 (Алгоритм Форни быстрого поиска значений ошибок)**.** $E_i = \frac{X_i^{-b}\Gamma(X_i^{-1})}{\prod_{j\neq i}(1-X_jX_i^{-1})}, 0 \le i < t$

24.3 Декодирование с помощью алгоритма Евклида

24.3.1 Расширенный алгоритм Евклида

Поиск наибольшего общего делителя $r_{-1}(x) = a(x), r_0(x) = b(x)$

$$r_{i-1}(x) = q_i(x)r_i(x) + r_{i+1}(x), \deg r_{i+1}(x) < \deg r_i(x)$$

НОД равен последнему ненулевому остатку $r_i(x)$

$$\begin{pmatrix} r_i(x) & r_{i-1}(x) \end{pmatrix} \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} r_{i+1}(x) & r_i(x) \end{pmatrix}$$

$$\begin{pmatrix} b(x) & a(x) \end{pmatrix} \underbrace{\prod_i \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix}}_{U(x)} = \begin{pmatrix} 0 & \gcd(a(x), b(x)) \end{pmatrix}$$

Теорема 24.2 (Безу)**.** Существуют многочлены $u(x), v(x) : b(x)u(x) + a(x)v(x) = \gcd(a(x), b(x))$

$$\text{Пусть } U_j(x) = \prod_{i=0}^{j} \begin{pmatrix} -q_i(x) & 1 \\ 1 & 0 \end{pmatrix} = U_{j-1} \underbrace{\begin{pmatrix} -q_j(x) & 1 \\ 1 & 0 \end{pmatrix}}_{Q_j(x)} = \begin{pmatrix} u_{j,0}(x) & u_{j-1,0}(x) \\ u_{j,1}(x) & u_{j-1,1}(x) \end{pmatrix}, U_{-1}(x) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} r_0(x) & r_{-1}(x) \end{pmatrix} \begin{pmatrix} u_{j,0}(x) & u_{j-1,0}(x) \\ u_{j,1}(x) & u_{j-1,1}(x) \end{pmatrix} = \begin{pmatrix} r_{j+1}(x) & r_j(x) \end{pmatrix}$$

1. $\deg u_{j,0}(x) = \deg u_{j-1,0}(x) + \deg q_j(x) = \sum_{i=0}^{j} \deg q_i(x) = \sum_{i=0}^{j} (\deg r_{i-1}(x) - \deg r_i(x)) = \deg r_{-1}(x) - \deg r_j(x)$
2. $u_{j,0}(x)u_{j-1,1}(x) - u_{j-1,0}(x)u_{j,1}(x) = \det U_j(x) = \prod_{i=0}^{j} \det Q_j(x) = (-1)^{j+1}$
3. $\gcd(u_{j,0}(x), u_{j,1}(x))$1. Если $f(x)|u_{j,0}(x), f(x)|u_{j,1}(x)$, то $f(x)|(u_{j,0}(x)u_{j-1,1}(x) - u_{j-1,0}(x)u_{j,1}(x))$
4. $r_{j+1}(x) = r_0(x)u_{j,0}(x) + r_{-1}(x)u_{j,1}(x)$
   $r_{j+1}(x) \equiv r_0(x)u_{j,0}(x) \mod r_{-1}(x)$ – похоже на ключевое уравнение
5. $\gcd(r_{j+1}(x), u_{j,0}(x)) = \gcd(r_{-1}(x), u_{j,0}(x))$
   $f(x)|r_{j+1}(x) \wedge f(x)|usj, 0(x) \implies f(x)|r_{-1}(x)\ f(x)|r_{-1}(x) \wedge f(xj|u_{j,0}(x) \implies f(x)|r_{j+1}(x)$

24.3.2 Алгоритм Сугиямы

Пусть $\delta = 2\tau + 1$

1. Пусть $r_{-1}(x) = x^{2\tau}, r_0(x) = S(x)$
2. Выполнять $r_{i-1}(x) = q_i(x)r_i(x) + r_{i+1}(x)$, пока не получится $\deg r_i(x) \ge \tau, \deg r_{i+1}(x) < \tau$
3. $\Gamma(x) = r_{i+1}(x), \Lambda(x) = u_{i,0}(x)$

Корректность алгоритма

1. Степени $r_i(x)$ монотонно убывают, т.е. условие останова достижимо
2. $\Gamma(x) = r_{i+1}(x) = r_0(x)u_{i,0}(x) + r_{-1}(x)u_{i,1}(x) = S(x)\Lambda(x) + x^{2\tau}u_{i,1}(x) \equiv S(x)\Lambda(x) \mod x^{2\tau}$
3. $\deg u_{i,0}(x) = \deg r_{-1}(x) - \deg r_i(x) \leq 2\tau - \tau \leq \tau$
4. Пусть $\Gamma'(x) \equiv S(x)\Lambda'(x) \mod x^{2\tau}, \deg\Lambda'(x) \leq \tau, \deg\Gamma'(x) < \tau$. Если $\Lambda'(x), \Gamma'(x)$ – истинные многочлены локаторов и значений ошибок, то $\gcd(\Lambda'(x), \Gamma'(x)) = 1$

$$\Gamma'(x)\Lambda(x) \equiv \Lambda(x)S(x)\Lambda'(x) \equiv \Gamma(x)\Lambda'(x) \mod x^{2\tau}$$

$\deg\Gamma'(x) + \deg\Lambda(x) < 2\tau, \deg\Gamma(x) + \deg\Lambda'(x) < 2\tau \implies \Gamma'(x)\Lambda(x) = \Gamma(x)\Lambda'(x)$ Из взаимной простоты $\Lambda'(x), \Gamma'(x)$ следует, что $\mu(x) = \frac{\Lambda(x)}{\Lambda'(x)} = \frac{\Gamma(x)}{\Gamma'(x)}$ – многочленю

$$\Gamma'(x)\mu(x) = \Gamma(x) = S(x)\Lambda(x) + x^{2\tau}u_{i,1}(x) = S(x)\Lambda'(x)\mu(x) + x^{2\tau}u_{i,1}(x)$$

$\implies \mu(x)|u_{i,1}(x)$. Но $\Lambda(x) = \mu(x)\Lambda'(x) = u_{i,0}(x)$ и $u_{i,1}(x)$ взаимно просты $\implies \mu(x) = const$

25 Альтернантные коды

25.1 Альтернантные коды

Теорема 25.1. $= (_0, \ldots, c_{n-1})$ – кодовое слово кода Рида-Соломона над $GF(q)$ в узком смысле ттогда, когда $c_i = f(\alpha_i), 0 \leq i < n$ (т.е. $c = ev(f)$), где $\deg f(x) < k, f(x) \in GF(q)[x]$

Определение. Альтернантным кодом длины n над полем $GF(q)$ называется код $\mathcal{A}(n, r, a, u)$ с проверочной матрицей

$$H = \begin{pmatrix} a_0^0 & a_1^0 & \ldots & a_{n-1}^0 \\ a_0^1 & a_1^1 & \ldots & a_{n-1}^1 \\ \vdots & \vdots & \ddots & \vdots \\ a_0^{r-1} & a_1^{r-1} & \ldots & a_{n-1}^{r-1} \end{pmatrix} (u_0, u_1, \ldots, u_{n-1}) = (H_{i,j})$$

где $a_i \in GF(q^m)$ – различные элементы, $u_i \in GF(q^m) \setminus 0$

25.2 Коды Гоппы

Определение. Пусть задан многочлен (многочлен Гоппы) $G(x) \in GF(q^m)[x]$ и $a_0, \ldots, a_{n-1} \in GF(q^m)$, причем $G(a_i) \neq 0$. Кодом Гоппы называется множество $(c_0, \ldots, c_{n-1}) \in GF(q)^n$.

$$\sum_{i=0}^{n-1} \frac{c_i}{x - a_i} \equiv 0 \mod G(x)$$

Утверждение. *Коды Гоппы являются альтернантными*

25.3 TODO Криптосистема Мак-Элиса

Доделать

26 Низкоплотностные коды

26.1 TODO Низкоплотностные коды

Доделать

26.2 TODO Основные характеристики

Доделать

26.3 TODO Конструкции низкоплотностных кодов

Доделать

27 Декодирование низкоплотностных кодов

27.1 TODO Декодирование низкоплотностных кодов

Доделать

28 Эволюция плотностей и порог итеративного декодирования низкоплотностных кодов

28.1 TODO Эволюция плотностей и порог итеративного декодирования низкоплотностных кодов

Доделать

29 Кодирование в стирающих каналах

29.1 TODO Кодирование в стирающих каналах

Доделать

29.2 TODO Цифровой фонтан

Доделать

29.3 TODO LT-коды и хищные коды

Доделать

30 Многоуровневые коды

30.1 TODO Битоперемеженная кодовая модуляция

Доделать

30.2 TODO Многоуровневые коды

Доделать