ТЕОРИЯ НА КОДИРАНЕТО

Иван Ланджев

16.01.2017

Съдържание

Глава 1

Основни понятия за линейни кодове

1.1 Комуникационни системи

Предаването на данни в комуникациоона система може да бъде проеедставено пространствено (радио, телеграф, телефон, телевизия и пр.) или времево (магнитна лента, плоча, филм, CD, дискове, флаш-памети и пр.). За да обхванем възможно най-широк кръг от системи, ние си представяме съставните им части като черни кутии, допускащи определен вход и изход; те получават определени данни, запазват ги, обработват ги и ги предават по-нататък. Една абстрактна комуникационна система се състои от следните части:

(1) източник на данни (message source);

(2) кодер на източника (source encoder);

(3) кодер на канала (channel encoder);

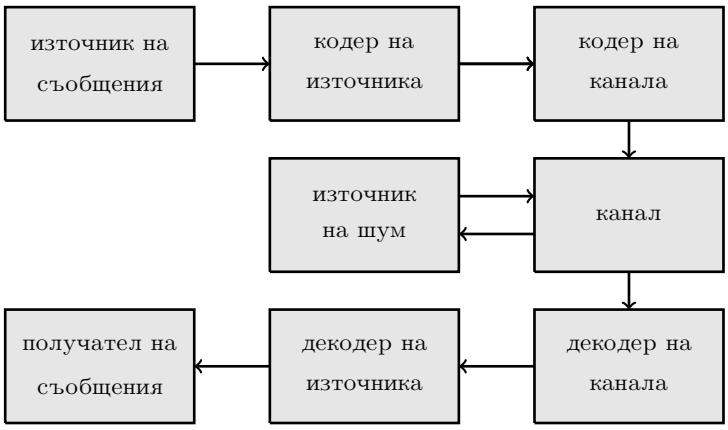
(4) канал (channel);

(5) източник на шум (noise);

(6) канален децодер (channel decoder);

(7) декодер на източника (source decoder);

(8) получател на съобщения (message sink)



Източникът на данни (1) е дискретен и стационарен. Той разполага с определен (краен) набор от съобщения. Отделните съобщения се избират чрез вероятностен процес с вероятности, които приемаме за известни. Този избор се осъществява независимо от конкретния момент. Съдържащата се в съобщенията информация се измерва чрез честотата им на появяване. Често срещащи се съобщения имат ниско информационно съдържание.

Кодерът на източника (2) преобразува съобщенията в такива сигнали, които могат да бъдат предавани от канала. Каналът може да приема за секунда определен брой сигнали. За да може да се осъществи относително висока скорост на предаване на данните кодерът съпоставя на относително често срещащите се съобщения такива изходни сигнали, които могат да бъдат предавани по-бързо. Функцията на кодера на източника се нарича компресиране на данните. При това съобщенията се освобождават от ненужния излишък. При предаването на данни единствената цел е запазване на информацията, съдържаща се в съобщението.

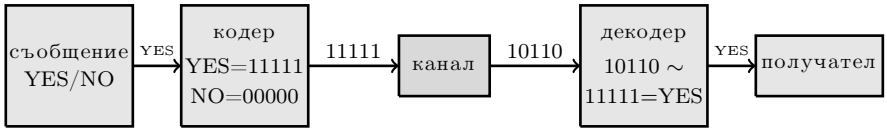
Каналът (4) е сърцето на информационната система и е независим от другите ѝ части. Така както един компютър може да бъде използван за различни цели, така и към един канал могат да бъдат закачени различни източници на данни. При наличието на източник на шум сигналите, предавани по канала, биват повредени със някаква статистически определена вероятност. Това води до загубата на известно количество информация; то се нарича еквивокация. Оставащата полезна част от информацията се нарича трансинформация. Източникът на шум не води само до загубата на информация; той може също да бъде разглеждан и като източник на съобщения. Причинените от шума смущения могат да бъдат разглеждани от своя страна като съобщения с определено инормационно съдържание. Каналът не се влияе от това, че потребителят не се интересува от тази информация. Той добавя тази ирелевантна информация, известна като ирелевантност, към трансинформацията. Така наблюдаваната на изхода информация произхожда само частично от източника; от нея трябва да бъде филтрирана трансинформацията. Под капацитет на канала разбираме съдържанието на трансинформация в един сигнал в средния случай, който се получава от оптимален източник на данни.

Задача на кодера на канала (3) е да гарантира в някаква степен надеждно предаване на съобщенията въпреки тези неблагоприятни обстоятелства. Кодерът групира входните сигнали в блокове от по k сигнала и добавя по $r = n - k$ контролни бита към всяка такава група. Отношението $R = k/n$ се нарича скорост. Теоремата на каналното кодиране на Шенон твърди, че при канал с шум с капацитет C за всяка фиксирана скорост $R < C$ данните могат да бъдат кодирани чрез код със скорост R така, че при предаването на данни да може да бъде достигнато произволно високо ниво на сигурност. За съжаление известните доказателства на тази теорема не дават указания как да бъдат конструирани такива кодове. Важна е и обратната теорема на каналното кодиране. При скорост на кода над капацитета на канала съществуват граници за надеждността на предаваните данни.

Декодерът на канала (6) запазва сигналите, получени от канала, в блокове от по n сигнала и се опитва да реконструира оригиналните k сигнала, обединени от кодера в блок. Когато това не му се удава казваме, че той прави грешка при декодиране. При чувствителни комуникационни системи грешките при декодиране могат да имат катастрофални последици.

Декодерът на източника (7) превежда предадените му сигнали от каналния декодер в такива сигнали, които могат да бъдат разбрани от потребителя.

Пример 1.1. Да разгледаме съвсем прост пример, в който единствените съобщения, които искаме да изпращаме са YES и NO.



В този случай са се появили две грешки и декодерът декодира получената дума 01001 в "най-близката" кодова дума, която е 00000 или YES. □

Пример 1.2. Множеството на всички имена на улици в произволен град може да се разглежда като код над 27-буквена азбука (включваща буквите от английската азбука и интервала като 27-ми символ). Обикновено това е пример за лошо кодиране. Така например в английския град Салфорд има улици с имена MILLFIELD DRIVE и HILLFIELD DRIVE. С такова кодиране не съществува възможност дори за откриване на единична грешка. □

Пример 1.3. Множеството на всички телефонни номера в Обединеното кралство (например) е 10-ичен код над 10-буквена азбука $\{0, 1, \ldots, 9\}$. За съжаление те се раздават по прозволен начин като с малко усилие те могат да бъдат генерирани така, че да се минимизира броя на погрешните свързвания. Възможно е да се построи код с над 82 милиона телефонни номера (които са повече от достатъчно за Великобритания), така че дори една цифра да бъде избрана погрешно да може да бъде осъществена правилната връзка.

Пример 1.4. The ISBN-code Всяка книга, издадена в последните години има международен стандартен книжен код (International Standard Book Number). Този номер е 10-буквена

кодова дума, определена от издателя. Например една книга може да има ISBN 0-19-859617-0. Тиретата може да се появяват на различни места и са всъщност несъществени.Първата цифра в думата показва езика, в случая английски. Следващите две цифри указват издателя, в случая Oxford University Press. Следващите шест цифри са номерът на книгата, определен от издателя и последната цифра се избира така, че цялото 10-цифрено число $x_1x_2\ldots x_{10}$ да удовлетворява

$$\sum_{i=1}^{10} ix_i \equiv 10 \pmod{11}. \tag{1.1}$$

Лявата страна на (1.1) се нарича претеглена проверочна сума за думата $x_1\ldots x_{10}$. При зададено 9-цифрено число $x_1\ldots x_9$ последната цифра x_{10} се избира така, че да е изпълнено $x_{10} \equiv \sum_{i=1}^{9} ix_i \pmod{11}$. Издателят трябва да използва и един специален символ X в последната позиция, ако $x_{10} = 10$. Например в 20th Century Chamber's Dictionary имаме следния ISBN:055010206-X. ISBN-кодът създаден за да открива (a) една грешка в произволна позиция; (b) всяка двойна грешка, получена от транспозиция на два символа.

(a) Да предположим, че полученият вектор $\boldsymbol{y} = y_1\ldots y_{10}$ съвпада със $\boldsymbol{x}$ във всички позиции освен в j-тата, която се получава като $x_j + a$, $a \neq 0$,. Тогава $\boldsymbol{y} = \sum iy_i = \sum ix_i + (ja) = ja \neq 0 \pmod{11}$ тъй като j и a са различни от 0.

(b) Нека $\boldsymbol{y}$ съвпада с $\boldsymbol{x}$ във всички позиции, освен в позиции j и k като x_j и x_k са разменили местата си. Тогава

$$\begin{aligned} y_{10} &= \sum_{i=1}^{9} y_i \\ &= \sum_{i=1}^{9} x_i + (k-j)x_j + (j-k)x_k \\ &= (k-j)(x_j - x_k) \not\equiv 0 \pmod{11}. \end{aligned}$$

ако $k \neq j$ И $x_j \neq x_k$. Да забележим, че тук съществено се използва идеята, че произведението на два ненулеви елемента от $\mathbb{Z}_{11}$ е различно от нула. Това не е вярно например в $\mathbb{Z}_{10}$, където $2 \cdot 5 = 0 \pmod{10}$, но $2 \neq 0$ и $5 \neq 0$. ISBN-кодът не може да поправи една грешка. □

Пример 1.5. Да допуснем, че генералният щаб HQ и корабът X имат идентични. Само HQ знае пътя през вражеската територия, по който X може да се завърне безопасно до HQ. HQ може да предава двоични данни до X и да изпрати маршрута $NNMNNWWSSWWNNNNW$. В тази ситуация надеждността на предаване на данните и по-важна от скоростта. Да разгледаме, как четирите съобщения могат да бъдат кодирани в двоични думи. Най-късият възможен код е следният:

$$C_1 : \begin{cases} 00 &= N \\ 01 &- W \\ 10 &= E \\ 11 &= S \end{cases}.$$

С други думи ние идентифицираме четирите съобщения N, W, E, S с четирите вектора от $\mathbb{F}_2^2$. Сега ще добавим излишък за да защитим тези съобщения от шум. Да разгледаме код C_2 с дължина 3, получен чрез добавяне на допълнителен символ както следва.

$$C_2 : \begin{cases} 000 & = & N \\ 011 & - & W \\ 101 & = & E \\ 110 & = & S \end{cases} .$$

Сега предването на съобщения е отнеме повече време, но ако в кодова дума се появи едн а грешка, то получената дума няма да е от кода и получателят ще разбере това. Така той би могъл да поиска съобщението да бъде изпратено отново. Така C_2 дава възможност за откриване на една грешка или, с други думи, това е код, откриващ една грешка.

Сега да предположим, че X може да получава съобщения, но не може да изисква повторно предаване насъобщенията, т.е. каналът е еднопосочен. Подобна е ситуацията, когато получаваме снимки от космоса или просвирваме запис от стара магнитна лента. В тези случаи е важно да извлечем колкото може повече информация от получените думи. Чрез подходящо добавяне на нови два символа към всяка дума от C_2 получваме нов код с дължина 5.

$$C_3 : \begin{cases} 00000 & = & N \\ 01101 & - & W \\ 10110 & = & E \\ 11011 & = & S \end{cases} .$$

Ако в коя да дума на C_3 се случи една грешка, ние ще можем не само да я забележим, но и да я поправим тъй като полученият вектор ще е "по-близо" до изпратената от всяка друга дума. Ако кодът се използва само за откриване на грешки, то той позволява откриването на две промени в коя да е кодова дума. □

1.2 Канали и канално кодиране

Целта на каналното кодиране е внасянето на излишък в предаваните данни, така че появяващите се грешки да могат да бъдат откривани и дори поправяни. В този раздел ще формализираме понятията откриване на грешки и поправяне на грешки. Ще въведем и някои естествени правила за декодиране, възстановяващи оригиналното съобщение и поправящи възникналите грешки.

Ще започнем с някои основни дефиниции.

Дефиниция 1.6. Нека $A = \{a_1, a_2, \ldots, a_q\}$ е множество с мощност q, което ще наричаме *кодова азбука* и чиито елементи ще наричаме *кодови символи*.

(i) Всяка редица $\boldsymbol{w} = w_1 w_2 \ldots w_n$, за която $w_i \in A$ за всяко i, наричаме *q-ична дума с дължина n над A*. Еквивалентно, ако A е поле, то $\boldsymbol{w}$ може да бъде разглеждана като вектор $(w_1, w_2, \ldots, w_n)$.

(ii) Всяко непразно множество C от q-ични думи с една и съща дължина n наричаме *q-ичен блоков код с дължина n*.

(iii) Всеки елемент на C наричаме *кодова дума от C*.

(iv) Броят на кодовите думи в C означаваме с $|C|$ и наричаме *мощност* на кода C.

(v) *Скорост на кода C* с дължина n наричаме величината $\log_q|C|/n$.

(vi) Код с дължина n и мощност M наричаме *(n, M)-код*.

Много често е удобно кодовата азбука A да има някаква структура, например да бъде крайно поле. Ако кодовата азбука е $A = \mathbb{F}_2 = \{0, 1\}$, то съответният код се нарича *двоичен код*. Примери за двоични кодове са:

- $C_1 = \{00, 01, 10, 11\}$ – $(2, 4)$ код;
- $C_2 = \{000, 011, 101, 110\}$ – $(3, 4)$-код;
- $C_3 = \{00000, 01101, 10110, 11011\}$ – $(3, 4)$-код.

Код над азбуката $\mathbb{F}_3 = \{0, 1, 2\}$ наричаме *троичен код*.

Дефиниция 1.7. Един *комуникационен канал* се състои от крайна канална азбука $A = \{a_1, a_2, \ldots, a_n\}$, както и от множество от вероятности $P(a_j|a_i)$, удовлетворяващи

$$\sum_{j=1}^{q} P(a_j|a_i) = 1,$$

за всички i. Тук $P(a_j|a_i)$ е условната вероятност да получим на изхода символа a_j при условие, че е изпратен символът a_i.

Дефиниция 1.8. Ще казваме, че един комуникационен канал е *без памет*, ако резултатът при предаването на един символ не зависи от резултата при предаване на предишните символи. С други думи за канал без памет, ако $\boldsymbol{c} = c_1c_2 \ldots c_n$ и $\boldsymbol{x} = x_1x_2 \ldots x_n$ са думи с дължина n, то

$$P(\boldsymbol{x}|\boldsymbol{c}) = \prod_{i=1}^{n} P(x_i|c_i).$$

Дефиниция 1.9. Канал без памет с канална азбука с мощност q, за който е изпълнено:

(i) всеки символ има една и съща вероятност $p < 1/2$ да бъде сгрешен,

(ii) ако един символ е сгрешен, то всяка от възможните $q - 1$ грешки е равновероятна

наричаме *q-ичен симетричен канал*.

По-специално *двоичен симетричен канал* е канал без памет с канална азбука $\{0,1\}$ и канални вероятности

$$\begin{aligned} P(1|0) = P(0|1) &= p, \\ P(0|0) = P(1|1) &= 1-p. \end{aligned}$$

Пример 1.10. Да предположим, че кодовите думи се избират от кода $\{000, 111\}$ и се изпращат по двоичен симетричен канал с вероятност за грешка $p = 0.05$. Да предположим, че получената дума е 110. Можем да се опитаме да пресметнем коя е по вероятната от двете кодове думи, пресмятайки вероятностите:

$$\begin{aligned} P(110|000) &= P(1|0)^2 \times P(0|0) \\ &= (0.05)^2(0.95) = 0.002375, \\ P(110|111) &= P(1|1)^2 \times P(0|1) \\ &= (0.95)^2(0.05) = 0.045125. \end{aligned}$$

Тъй като втората вероятност е по-голяма, можем да заключим, че е по-вероятно да е била изпратена думата 111.

Нека е даден комуникационен канал и код за защита от грешки. По канала се изпращат само кодови думи. Да предположим, че на изхода е получена думата $\boldsymbol{w}$. Ако $\boldsymbol{w}$ е валидна кодова дума, то можем да заключим, че при предаването не са възникнали грешки. В противен случай знаем, че има грешки и се нуждаем от правило, по което да определим коя е най-вероятната кодова дума. Такова правило се нарича *правило за декодиране*. В следващите раздели ще представим две такива правила.

1.3 Декодиране по приципа на максималното правдоподобие

Да предположим, че по комуникационен канал се изпращат думи от кода C. Ако на изхода получим дума $\boldsymbol{x}$, бихме могли да пресметнем вероятността $P(\boldsymbol{x}|\boldsymbol{c})$ (вероятността да се получи $\boldsymbol{x}$ при условие, че е изпратена кодовата дума $\boldsymbol{c}$) за всички $\boldsymbol{c} \in C$. Правилото за декодиране по принципа на максималното правдоподобие се състои в това да се декодира в кодовата дума $\boldsymbol{c}_{\boldsymbol{x}}$, ако

$$P(\boldsymbol{x}|\boldsymbol{c}_{\boldsymbol{c}_x}) = \max_{\boldsymbol{c} \in C} P(\boldsymbol{x}|\boldsymbol{c}).$$

Разглеждат се два вида декодиране по принципа на максималното правдоподобие:

(1) *Пълно декодиране по принципа на максималното правдоподобие.* Ако е получена дума $\boldsymbol{x}$, намираме най-вероятната кодова дума; ако съществува повече от една такава, избираме по случаен начин една от тях.

(2) *Непълно декодиране по принципа на максималното правдоподобие.* Ако е получена дума $\boldsymbol{x}$, намираме най-вероятната кодова дума; ако съществува повече от една такава, изискваме повторно предаване на съобщението.

1.4 Разстояние на Хеминг

Нека думи от двоичния код C се изпращат по двоичен симетричен канал с вероятност за грешка $p < 1/2$ (на практика p е винаги много по-малко от $1/2$). Ако получената на изхода дума е $\boldsymbol{x}$, то за всяка кодова дума $\boldsymbol{c} \in C$ имаме

$$P(\boldsymbol{x}|\boldsymbol{c}) = p^e(1-p)^e,$$

където n е дължината на кода (и на $\boldsymbol{x}$), а e е броят на позициите, в които $\boldsymbol{x}$ и $\boldsymbol{c}$ се различават. Тъй като $p < 1/2$, имаме $1 - p > p$ и горната вероятност е по-голяма за по-големи стойности на $n - e$, т.е. за по-малки стойности на e. Следователно горната вероятност се максимизира за такива думи $\boldsymbol{c}$, за които e е минимално. Това разсъждение ни води до фундаменталното понятие *разстояние на Хеминг.*

Дефиниция 1.11. Нека $\boldsymbol{x}$ и $\boldsymbol{y}$ са думи с дължина n над азбуката A. *Разстояние на Хеминг* от $\boldsymbol{x}$ до $\boldsymbol{y}$, което означаваме с $d(\boldsymbol{x}, \boldsymbol{y})$, наричаме броя на позициите, в които $\boldsymbol{x}$ и $\boldsymbol{y}$ се различават. Ако $\boldsymbol{x} = x_1 \dots x_n$ и $\boldsymbol{y} = y_1 \dots y_n$, то

$$d(\boldsymbol{x}, \boldsymbol{y}) = d(x_1, y_1) + \dots + d(x_n, y_n), \tag{1.2}$$

където x_i и y_i се разглеждат като думи с дължина 1 и

$$d(x_i, y_i) = \begin{cases} 1 & \text{ако } x_i \neq y_i, \\ 0 & \text{ако } x_i = y_i. \end{cases}$$

Пример 1.12. (i) Нека $A = \{0, 1\}$ и $\boldsymbol{x} = 01010$, $\boldsymbol{y} = 01101$, $\boldsymbol{z} = 11101$. Тогава

$$d(\boldsymbol{x}, \boldsymbol{y}) = 3,\ d(\boldsymbol{y}, \boldsymbol{z}) = 1,\ d(\boldsymbol{z}, \boldsymbol{x}) = 4.$$

(ii) Нека $A = \{0, 1, 2, 3, 4\}$ и $\boldsymbol{x} = 1234$, $\boldsymbol{y} = 1423$, $\boldsymbol{z} = 3214$. Тогава

$$d(\boldsymbol{x}, \boldsymbol{y}) = 3,\ d(\boldsymbol{y}, \boldsymbol{z}) = 4,\ d(\boldsymbol{z}, \boldsymbol{x}) = 2.$$

□

Теорема 1.13. Нека $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ са думи с дължина n над азбуката A. Тогава имаме

(i) $0 \leq d(\boldsymbol{x}, \boldsymbol{y}) \leq n$;

(ii) $d(\boldsymbol{x}, \boldsymbol{y}) = 0$ тогава и само тогава, когато $\boldsymbol{x} = \boldsymbol{y}$;

(iii) $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$;

(iv) (неравенство на триъгълника) $d(\boldsymbol{x}, \boldsymbol{z}) \leq d(\boldsymbol{x}, \boldsymbol{y}) + d(\boldsymbol{y}, \boldsymbol{z})$.

Доказателство. (i), (ii) и (iii) следват непосредствено от дефиницията на разстояние на Хеминг. От (1.2) следва, че е достатъчно да докажем (iv) за случая $n = 1$. Сега ако $\boldsymbol{x} = \boldsymbol{z}$, $d(\boldsymbol{x}, \boldsymbol{z}) = 0$ и (iv) очевидно е вярно. Ако $\boldsymbol{x} \neq \boldsymbol{z}$, то или $\boldsymbol{y} \neq \boldsymbol{x}$ или $\boldsymbol{y} \neq \boldsymbol{z}$ и (iv) отново е изпълнено. □

1.5 Декодиране в най-близкия съсед

Нека по комуникационен канал се изпращат думи от кода C и нека на изхода е получена дума $\boldsymbol{x}$. *Декодиране в най-близкия съсед* (или *декодиране по минимално разстояние*) е такова правило, което декодира $\boldsymbol{x}$ във $\boldsymbol{c_x}$, ако $d(\boldsymbol{x}, \boldsymbol{c_x})$ е минимално по всички думи на C, т.е.

$$d(\boldsymbol{x}, \boldsymbol{c_x}) = \min_{\boldsymbol{c} \in C} d(\boldsymbol{x}, \boldsymbol{c}).$$

Както и в случая на декодиране по принципа на максималното правдоподобие, и тук различаваме пълно и непълно декодиране. Ако при получена дума $\boldsymbol{x}$ съществуват две или повече кодови думи $\boldsymbol{c_x}$, за които се достига минимум, то правилото за пълно декодиране избира по произволен начин една от тях, докато непълното декодиране изисква повторно предаване.

Теорема 1.14. За двоичен симетричен канал с вероятност за грешка $p < 1/2$ декодирането по принципа на максималното правдоподобие съвпада с декодиране в най-близкия съсед.

Доказателство. Нека C е използваният код, а $\boldsymbol{x}$ е получената дума (с дължина n). За всеки вектор $\boldsymbol{c}$ с дължина n и за всяко $0 \leq i \leq n$

$$d(\boldsymbol{x}, \boldsymbol{c}) = i \Longleftrightarrow P(\boldsymbol{x}|\boldsymbol{c}) = p^i(1-p)^{n-i}.$$

Тъй като $p < 1/2$, имаме

$$p^0(1-p)^n > p^1(1-p)^{n-1} > p^2(1-p)^{n-2} > \ldots > p^n(1-p)^0.$$

По дефиниция, декодирането по принципа на максималното правдоподобие декодира $\boldsymbol{x}$ във $\boldsymbol{c}$, ако вероятността $P(\boldsymbol{x}|\boldsymbol{c})$ е максимална. Това се случва точно когато $d(\boldsymbol{x}, \boldsymbol{c})$ е минимално. Следователно то е същото като декодиране в най-близкия съсед. □

Пример 1.15. Нека думите от кода

$$C = \{0000, 0011, 1000, 1100, 0001, 1001\}$$

се изпращат по двоичен симетричен канал. Да предположим, че получената дума е $\boldsymbol{x} = 0111$. Тогава

$$\begin{aligned} d(0111, 0000) &= 3, \\ d(0111, 0011) &= 1, \\ d(0111, 1000) &= 4, \\ d(0111, 1100) &= 3, \\ d(0111, 0001) &= 2, \\ d(0111, 1001) &= 3. \end{aligned}$$

Използвайки правилото за декодиране в най-близкия съсед, декодираме $\boldsymbol{x}$ в 0011. □

Пример 1.16. Нека $C = \{000, 011\}$ е двоичен код. Непълно декодиране в най-близкия съсед е показано в таблицата по-долу. Символът "–" показва, че искаме повторно предаване на кодовата дума.

$\boldsymbol{x}$	$d(\boldsymbol{x}, 000)$	$d(\boldsymbol{x}, 011)$	$\boldsymbol{c_x}$
000	0	2	000
100	1	3	000
010	1	1	–
001	1	1	–
110	2	2	–
101	2	2	–
011	2	0	011
111	3	1	011

□

1.6 Минимално разстояние на код

Една от основните характеристики на един код е неговото минимално разстояние.

Дефиниция 1.17. *Минимално разстояние* на код C, съдържащ поне две кодови думи, наричаме най-малкото разстояние между две различни думи от кода. Минималното раз стояние на C означаваме с $d(C)$, т.е.

$$d(C) = \min\{d(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x}, \boldsymbol{y} \in C, \boldsymbol{x} \neq \boldsymbol{y}\}.$$

Дефиниция 1.18. Код с дължина n, мощност M и минимално разстояние d наричаме (n, M, d)-*код*. Числата n, M, d наричаме *параметри* на кода.

Пример 1.19. (i) Нека $C = \{00000, 00111, 11111\}$ е двоичен код. Тогава $d(C) = 2$, тъй като

$$\begin{aligned} d(00000, 00111) &= 3, \\ d(00000, 11111) &= 5, \\ d(00111, 11111) &= 2. \end{aligned}$$

Следователно, C е $(5, 3, 2)$-код.

(ii) Нека $C = \{000000, 000111, 111222\}$ е троичен код, т.е. код над азбуката $A = \{0, 1, 2\}$. Тогава $d(C) = 3$, тъй като

$$\begin{aligned} d(000000, 000111) &= 3, \\ d(000000, 111222) &= 6, \\ d(000111, 111222) &= 6. \end{aligned}$$

и C е троичен $(6, 3, 3)$-код. □

Дефиниция 1.20. Нека u е цяло положително число. Казваме, че кодът C *открива* u *грешки*, ако резултатът от появяването на не повече от u грешки в коя да е кодова дума е дума, която не се съдържа в кода C. Казваме, че C открива точно u грешки, ако той открива u, но не открива $u = 1$ грешки.

Пример 1.21. (i) Двоичният код $C = \{00000, 00111, 11111\}$ открива една грешка, тъй като промяната на един бит в коя да е позиция на произволна кодова дума не води до кодова дума. С други думи

$$\begin{aligned} 00000 &\rightarrow 00111 \text{ нуждаем се от промяна в три позиции,} \\ 00000 &\rightarrow 11111 \text{ нуждаем се от промяна в пет позиции,} \\ 00111 &\rightarrow 11111 \text{ нуждаем се от промяна в две позиции.} \end{aligned}$$

Очевидно C открива точно една грешка, тъй като промяната на символите в първите две позиции на 00111 води друга кодова дума 11111, т.е. C не е код откриващ две грешки.

(i) Троичният код $C = \{000000, 000111, 111222\}$ открива две грешки, тъй като промяната на един или два символа в кои да е позиции на произволна кодова дума не води до кодова дума:

$$\begin{aligned} 000000 &\rightarrow 000111 \text{ нуждаем се от промяна в три позиции,} \\ 000000 &\rightarrow 111222 \text{ нуждаем се от промяна в шест позиции,} \\ 000111 &\rightarrow 111222 \text{ нуждаем се от промяна в шест позиции.} \end{aligned}$$

Кодът C открива точно две грешки, тъй като промяната на символите в последните три позиции на 000000 води кодовата дума 000111, т.е. C не е код откриващ три грешки. □

Теорема 1.22. Един код C открива u грешки тогава и само тогава, когато $d(C) \geq u+1$, т.е код с минимално разсояние d открива точно $d-1$ грешки.

Доказателство. Да предположим, че $d(c) \geq u+1$. Ако $\boldsymbol{c}$ и $\boldsymbol{x}$ са такива, че $1 \leq d(\boldsymbol{c}, \boldsymbol{x}) \leq u < d$, то $\boldsymbol{x} \notin C$. Следователно C открива u грешки.

От друга страна, ако $d(C) < u+1$ съществуват такива думи $\boldsymbol{c}_1$, $\boldsymbol{c}_2 \in C$, за които $1 \leq d(\boldsymbol{c}_1, \boldsymbol{c}_2) = d(C) \leq u$. Следователно е възможно, започвайки от $\boldsymbol{c}_1$ и променяйки $d(C)$ символа да получим кодовата дума $\boldsymbol{c}_2$. Следователно C не е код откриващ u грешки. □

Дефиниция 1.23. Нека v е цяло положително число. Казваме, че кодът C *поправя* v *грешки*, ако непълно декодиране в най-близкия съсед поправя v или по-малко грешки. Казваме, че C *поправя точно* v *грешки*, ако C поправя v, но не поправя $v+1$ грешки.

Пример 1.24. Да разгледаме двоичния код $C = \{000, 111\}$. Използвайки правилото за декодиране в най-близкия съсед получаваме, че

- ако изпратената дума е 000 и се е случила една грешка, то получената дума е 100, 010 или 001 и тя ще бъде декодирана като 000;

- ако изпратената дума е 111 и се е случила една грешка, то получената дума е 110, 101 или 011 и тя ще бъде декодирана като 111.

Във всички случаи грешката ще бъде поправена. Следователно C поправя една грешка.

Ако са се случили поне две грешки, правилото за декодиране може да даде грешна дума. Например, ако е изпратена думата 000, а е получена 011, то последната ще се декодира като 111 като използваме правилото за декодиране в най-близкия съсед. Следователно C поправя точно една грешка. □

Теорема 1.25. Кодът C поправя v грешки тогава и само тогава, когато $d(C) \geq 2v+1$, т.е. код с минимално разстояние d е код поправящ точно $\lfloor (d-1)/2 \rfloor$ грешки. (Тук $\lfloor x \rfloor$ е най-голямото цяло число, по-малко или равно на x.)

Доказателство. ($\Leftarrow$) Да предположим, че $d(C) \geq 2v+1$. Нека $\boldsymbol{c}$ е изпратената кодова дума, а $\boldsymbol{x}$ е получената дума. Ако при предаването са се случили не повече от v грешки, то $d(\boldsymbol{x}, \boldsymbol{x}) \leq v$. Следователно за всяка кодова дума $\boldsymbol{c}' \in C$, $\boldsymbol{c}' \neq \boldsymbol{c}$ имаме

$$\begin{aligned} d(\boldsymbol{x}, \boldsymbol{c}') &\geq d(\boldsymbol{c}, \boldsymbol{c}') - d(\boldsymbol{x}, \boldsymbol{c}) \\ &\geq 2v+1-v \\ &= v+1 \\ &< d(\boldsymbol{x}, \boldsymbol{c}). \end{aligned}$$

Така $\boldsymbol{x}$ ще бъде декодирана (коректно) във $\boldsymbol{c}$, ако се използва правилото за декодиране в най-близкия съсед. Това показва, че C поправя v грешки.

($\Rightarrow$) Нека C поправя v грешки. Ако $d(c) < 2v+1$, то съществуват два различни думи $\boldsymbol{c}, \boldsymbol{c}' \in C$, за които $d(\boldsymbol{c}, \boldsymbol{c}') = d(C) \leq 2v$. Ще покажем, че ако изпратената дума е $\boldsymbol{c}$ и са станали не повече от v грешки, то може да се случи така, че правилото за декодиране в най-близкия съсед да декодира неправилно в $\boldsymbol{c}'$ или да не може въобще да декодира (поради нееднозначност на най-близката дума). Това ще противоречи на допускането, че C поправя v грешки, откъдето $d(C) \geq 2v+1$.

Да забележим, че ако $d(\boldsymbol{c}, \boldsymbol{c}') < v+1$, то $\boldsymbol{c}$ може да се трансформира във $\boldsymbol{c}'$ чрез не повече от v промени на символи. Тези грешки няма да могат да се поправят (и дори да се открият!), тъй като $\boldsymbol{c}'$ е отново в C. Това би противоречало на допускането, че C поправя v грешки. Следователно, $d(\boldsymbol{c}, \boldsymbol{c}') \geq v+1$. Без ограничение на общността можем да предположим, че $\boldsymbol{c}$ и $\boldsymbol{c}'$ се различават в първите $d = d(C)$ позиции, където $d+1 \leq d \leq 2v$. Ако е получена думата

$$\boldsymbol{x} = \underbrace{x_1 x_2, \ldots x_v}_{\text{съвпада със } \boldsymbol{c}'} \underbrace{x_{v+1}, x_{v+2}, \ldots x_d}_{\text{съвпада със } \boldsymbol{c}} \underbrace{x_{d+1}, x_{d+2}, \ldots x_n}_{\text{съвпада и с двете}}.$$

то имаме

$$d(\boldsymbol{x}, \boldsymbol{c}') = d - v \leq v = d(\boldsymbol{x}, \boldsymbol{c}).$$

Следователно имаме, че $d(\boldsymbol{x}, \boldsymbol{c}') < d(\boldsymbol{x}, \boldsymbol{c})$, в който случай $\boldsymbol{x}$ ще се декодира неправилно в $\boldsymbol{c}'$, или $d(\boldsymbol{x}, \boldsymbol{c}') = d(\boldsymbol{x}, \boldsymbol{c})$ и ще имаме неопределеност. □

1.7 Задачи

1. Suppose that codewords from the binary code $\{000, 100, 111\}$ are being sent over a binary symmetric channel with crossover probability $p = 0.03$. Use the maximum likelihood decoding rule to decode the following received words:

   (a) 010; (b) 011; (c) 001.

2. Consider a memoryless binary channel with channel probabilities

   $$\Pr(0 \text{ received } |0 \text{ sent }) = 0.7, \ \ \Pr(1 \text{ received } |1 \text{ sent }) = 0.8.$$

   If codewords from the code $\{000, 100, 111\}$ are being sent over this channel, use the maximum likelihood decoding rule to decode the follwing received words:

   (a) 010; (b) 011; (c) 001.

3. Let $C = \{001, 011\}$ be a binary code.

   (a) Suppose we have a memoryless binary channel with the following probabilities

   $$\Pr(0 \text{ received } |0 \text{ sent }) = 0.1, \ \ \Pr(1 \text{ received } |1 \text{ sent }) = 0.5.$$

   Use the maximum likelihood decoding rule to decode the received word 000.

   (b) Use the neares neighbour decoding rule to decode 000.

4. For the binary code $C = \{01101, 00011, 10110, 11000\}$, use the nearest neighbour decoding rule to decode the following received words:

   (a) 00000; (b) 01111; (c) 10110; (d) 10011; (e) 11011.

5. For the ternary code $C = \{00122, 12201, 20110, 22000\}$, use the nearest neighbour decoding rule to decode the following received words:

   (a) 01122; (b) 10021; (c) 22022; (d) 20120.

6. Construct the incomplete maximum likelihood decoding table for each of the following binary codes:

   (a) $C = \{101, 111, 011\}$;

   (b) $C = \{000, 001, 010, 011\}$.

7. Determine the number of binary codes with parameters $(n, 2, n)$ for $n \geq 2$.

Глава 2

Линейни кодове

2.1 Векторни пространства над крайни полета

Ще започнем с някои дефиниции и факти за векторни пространства над крайни полета, които са добре известни.

Дефиниция 2.1. Нека $\mathbb{F}_q$ е крайно поле от ред q. Едно непразно множество V, в което са зададени операциите събиране и умножение по скалар от $\mathbb{F}_q$ наричаме *векторно пространство* над $\mathbb{F}_q$ ако за всички $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in V$ и всички $\lambda, \mu \in \mathbb{F}_q$ са в сила свойствата:

(*i*) $\boldsymbol{u} + \boldsymbol{v} \in V$;

(*ii*) $(\boldsymbol{u} + \boldsymbol{v}) + \boldsymbol{w} = \boldsymbol{u} + (\boldsymbol{v} + \boldsymbol{w})$;

(*iii*) съществува елемент $\mathbf{0} \in V$ със свойството $\mathbf{0} + \boldsymbol{v} = \boldsymbol{v} = \boldsymbol{v} + \mathbf{0}$ за всички $\boldsymbol{v} \in V$;

(*iv*) за всяко $\boldsymbol{u} \in V$ съществува елемент от V, който означаваме с $-\boldsymbol{u}$ такъв, че $\boldsymbol{u} + (-\boldsymbol{u}) = \mathbf{0} = (-\boldsymbol{u} + \boldsymbol{u}$;

(*v*) $\boldsymbol{u} + \boldsymbol{v} = \boldsymbol{v} + \boldsymbol{u}$;

(*vi*) $\lambda \boldsymbol{v} \in V$;

(*vii*) $\lambda(\boldsymbol{u} + \boldsymbol{v}) = \lambda \boldsymbol{u} + \lambda \boldsymbol{v}$, $(\lambda + \mu)\boldsymbol{u} = \lambda \boldsymbol{u} + \mu \boldsymbol{u}$;

(*viii*) $(\lambda\mu)\boldsymbol{u} = \lambda(\mu \boldsymbol{u})$;

(*ix*) ако 1 е мултипликативната единица на $\mathbb{F}_q$, то $1\boldsymbol{u} = \boldsymbol{u}$.

Нека $\mathbb{F}_q^n$ е множеството на наредените n-орки с компоненти във $\mathbb{F}_q$:

$$\mathbb{F}_q^n = \{(v_1, \ldots, v_n) \mid v_i \in \mathbb{F}_q\}.$$

Дефинираме събиране и умножение по скаларот $\mathbb{F}_q^n$ покомпонентно, използвайки събирането и умножението във $\mathbb{F}_q$, т.е. ако $\lambda \in \mathbb{F}_q$ и

$$\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}_q^n, \boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{F}_q^n,$$

то

$$\boldsymbol{v} + \boldsymbol{w} = (v_1 + w_1, \ldots, v_n + w_n) \in \mathbb{F}_q^n,$$

$$\lambda \boldsymbol{v} = (\lambda v_1, \ldots, \lambda v_n) \in \mathbb{F}_q^n.$$

С $\mathbf{0}$ ще означаваме нулевия вектор $(0, \ldots, 0) \in \mathbb{F}_q^n$.

Пример 2.2. Множествата по-долу са примери за векторни пространства:

(*i*) $V_1 = \mathbb{F}_q^n$ за всяка степен на просто q;

(*ii*) $V_2 = \{\mathbf{0}\}$;

(*iii*) $V_3 = \{(\lambda, \ldots, \lambda) \mid \lambda \in \mathbb{F}_q\}$ за всяка степен на просто q;

(*iv*) за $q = 2$, $V_4 = \{(0,0,0,0), (1,1,1,0), (0,1,1,1), (1,0,0,1)\}$;

(*v*) за $q = 3$, $V_5 = \{(0,0,0), (0,1,2), (0,2,10\}$.

Дефиниция 2.3. Едно непразно подмножество C на векторното пространство V наричаме *подпространст во на* V, ако самото C е векторно пространство със същото събиране и умножение по скалар както във V.

Лесно се проверява, че в Пример 2.2 V_2 е подпространство на V_1 и V_3, V_3 еподпространство на V_1, V_4 е подпространство на $\mathbb{F}_2^4$ и V_5 е подпространство на $\mathbb{F}_3^3$.

Теорема 2.4. Едно непразно подмножество C на векторното пространство V над $\mathbb{F}_q$ е подпространство тогава и само тогава, когато за всекидва вектора $\boldsymbol{x}, \boldsymbol{y} \in C$ и за всъеки два елемента $\lambda, \mu \in \mathbb{F}_q$, $\lambda \boldsymbol{x} + \mu \boldsymbol{y} \in C$.

Доказателството на тази теорема е очевидно. За $q = 2$ това необходимо и достатъчно условие за C приема следния по прост вид: ако V е векторно пространство над $\mathbb{F}_2$, то $C \subseteq V$ е подпространство тогава и само тогава, когато за всеки два вектора $\boldsymbol{x}, \boldsymbol{y} \in C$ имаме $\boldsymbol{x} + \boldsymbol{y} \in C$.

Дефиниция 2.5. Let V be a vector space over $\mathbb{F}_q$. A *linear combination* of $\boldsymbol{v}_1$, …, $\boldsymbol{v}_r \in V$ is a vector of the form $\lambda_1 \boldsymbol{v}_1 + \ldots + \lambda_r \boldsymbol{v}_r$, where $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$ are some scalars.

Дефиниция 2.6. Нек V е векторно пространство над $\mathbb{F}_q$. Едно множество от вектори $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r\}$ във V наричаме *линейно независимо*, ако от

$$\lambda_1 \boldsymbol{v}_1 + \ldots + \lambda_r \boldsymbol{v}_r = \mathbf{0}$$

следва $\lambda_1 = \ldots = \lambda_r = 0$. Едно множество *линейно зависимо*, ако то не е линейно независимо, т.е. съществуват скалари $\lambda_1, \ldots, \lambda_r \in \mathbb{F}_q$, не всички равни на нула, за които $\lambda_1 \boldsymbol{v}_1 + \ldots + \lambda_r \boldsymbol{v}_r = \mathbf{0}$.

Всяко множество, съдържащо вектора $\mathbf{0}$ е линейно зависимо.

Дефиниция 2.7. Нека V е векторно пространство над $\mathbb{F}_q$ и нека $S = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ е непразно подмножество на V. *Линейна обвивка* (или само *обвивка*) на S дефинираме като множеството на всички линейни комбинации на векторите от S:

$$\langle S\rangle = \{\lambda_1\boldsymbol{v}_1 + \ldots + \lambda_k\boldsymbol{v}_k \mid \lambda_i \in \mathbb{F}_q\}.$$

Ако $S = \emptyset$, то полагаме $\langle S\rangle = \{\mathbf{0}\}$. Лесно се проверява, че $\langle S\rangle$ е подпространство на V. Ще казваме, че това подпространство е *породено* от S. Нека е дадено подпространството C на V. Едно подмножество S на C наричаме *пораждащо множество на* C, ако $C = \langle S\rangle$.

Пример 2.8. (*i*) Ако $q = 2$ и $S = (0001, 0010, 0100)$, то

$$\langle S\rangle = \{0000, 0001, 0010, 0100, 0011, 0101, 0110, 0111\}.$$

(*ii*) Ако $q = 2$ и $S = \{0001, 1000, 1001\}$, то

$$\langle S\rangle = \{0000, 0001, 1000, 1001\}.$$

(*iii*) Ако $q = 3$ и $S = \{0001, 1000, 1001\}$, то

$$\langle S\rangle = \{0000, 0001, 0002, 1000, 2000, 1001, 1002, 2001, 2002\}.$$

Дефиниция 2.9. Нека V е векторно пространство над $\mathbb{F}_q$. Едно непразно подмножество $B = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ на V наричаме *базис* на V, ако $V = \langle B\rangle$ и B е линейно независимо.

Ако $B = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ е базис на V, то всеки вектор $\boldsymbol{v} \in V$ може да бъде представен по единствен анчин като линейна комбинация на вектори от B. Едно векторно пространство над крайно поле $\mathbb{F}_q$ (а всъщност и над произволно поле) може да има много базиси. Всички те съдържат един и същ брой вектори. Този брой наричаме *размерност на* V над $\mathbb{F}_q$ и означаваме с $\dim V$. В случаите, когато V може да се разглежда като векторно пространство над повече от едно поле, ще използваме означението $\dim_{\mathbb{F}_q} V$ за да избегнем двусмисленост.

Теорема 2.10. Нека V е векторно пространство над $\mathbb{F}_q$. Ако $\dim V = k$, то

(*i*) V има q^k елементи;

(*ii*) V има $\frac{1}{k!}\prod_{i=0}^{k-1}(q^k - q^i)$ различни базиси.

Доказателство. (*i*) Ако $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ е базис за V, то

$$V = \{\lambda_1\boldsymbol{v}_1 + \ldots + \lambda_k\boldsymbol{v}_k \mid \lambda_1, \ldots, \lambda_k \in \mathbb{F}_q\}.$$

Тъй като за всеки от елементите λ_i имаме точно q възможности и тъй като всеки вектор се представя по единствен начин като линейна комбинация на базисни вектори, получаваме, че V има точно q^k елементи.

(*ii*) Нека $B = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ е базис за V. Тъй като $\boldsymbol{v}_1 \neq \mathbf{0}$, то съществуват $q^k - 1$ възможности за избор на $\boldsymbol{v}_1$. Тъй като B е басис, то еизпълнено $\boldsymbol{v}_2 \notin \langle \boldsymbol{v}_1\rangle$, откъдето

следва, че съществуват $q^k - q$ възможни избора за $\boldsymbol{v}_2$. Въобще за всяко i, $2 \leq i \leq k$, имаме $\boldsymbol{v}_i \notin \langle \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1} \rangle$ и така имаме $q^k - q^i$ възможности за $\boldsymbol{v}_i$. Следователно броят на наредените базиси $v_1, \ldots, v_k$ е равен на $\prod_{i=0}^{k-1}(q^k - q^i)$ Тъй като редът на векторите в базиса е несъществен, то броят на различните (ненаредени) базиси на V е

$$\frac{1}{k!}\prod_{i=0}^{k-1}(q^k - q^i).$$

□

Пример 2.11. Нека $q = 2$, $S = \{0001, 0010, 0100\}$ и $V = \langle S \rangle$. Тогава

$$V = \{0000, 0001, 0010, 0100, 0011, 0101, 0110, 0111\}.$$

Да отбележим, че векторите в S са линейно независими и така $\dim V = 3$. Съгласно Теорема 2.10 броят на различните базиси за V е

$$\frac{1}{k!}\prod_{i=0}^{k-1}(2^k - 2^i) = \frac{1}{3!}(2^3 - 1)(2^3 - 2)(2^3 - 2^2) = 28.$$

Дефиниция 2.12. Нека $\boldsymbol{v} = (v_1, \ldots, v_n), \boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{F}_q^n$. *Скаларно произведение* (още *Евклидово вътрешно произведение*) на векторите $\boldsymbol{v}$ и $\boldsymbol{w}$ дефинираме чрез

$$\boldsymbol{v} \cdot \boldsymbol{w} = v_1 w_1 + \ldots v_n w_n \in \mathbb{F}_q.$$

Ще казваме, че векторите $\boldsymbol{v}$ и $\boldsymbol{w}$ са *ортогонални*, ако $\boldsymbol{v} \cdot \boldsymbol{w} = 0$. Нека S е непразно подмножество на $\mathbb{F}_q^n$. *Ортогонално допълнение* $S^\perp$ на S наричаме множеството

$$S^\perp = \{\boldsymbol{v} \in \mathbb{F}_q^n \mid \boldsymbol{v} \cdot \boldsymbol{s} = 0 \text{ за всяко } \boldsymbol{s} \in S\}.$$

Ако $S = \emptyset$, то имаме $S^\perp = \mathbb{F}_q^n$.

Лесно се проверява, че $S^\perp$ е подпространство на $\mathbb{F}_q^n$ за всяко подмножество S от вектори от $\mathbb{F}_q^n$, както и че $\langle S \rangle^\perp = S^\perp$.

Скаларното произведение е пример за вътрешно произведение в $\mathbb{F}_q^n$. *Вътрешно произведение* в $\mathbb{F}_q^n$ наричаме всяко сдвояване $\langle , \rangle : \mathbb{F}_q^n \times \mathbb{F}_) q^n \to \mathbb{F}_q^n$, удволетворяващо условията: за всички вектори $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathbb{F}_q^n$ е в сила

(*a*) $\langle \boldsymbol{u} + \boldsymbol{v}, \boldsymbol{w} \rangle = \langle \boldsymbol{u}, \boldsymbol{w} \rangle + \langle \boldsymbol{v}, \boldsymbol{w} \rangle$;

(*b*) $\langle \boldsymbol{u}, \boldsymbol{v} + \boldsymbol{w} \rangle = \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \langle \boldsymbol{u}, \boldsymbol{w} \rangle$;

(*c*) $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$ за всички $\boldsymbol{u} \in \mathbb{F}_q^n$ тогава и само тогава, когато $\boldsymbol{v} = \boldsymbol{0}$;

(*d*) $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 0$ за всички $\boldsymbol{v} \in \mathbb{F}_q^n$ тогава и само тогава, когато $\boldsymbol{u} = \boldsymbol{0}$.

В теория на кодирането се използват и други вътрешни произведения, като Ермитово произведение или симплектично произведение, които са различни от Евклидовото.

Пример 2.13. (*i*) Нека $q = 2$ и нека $n = 4$. Ако $\boldsymbol{u} = (1,1,1,1)$, $\boldsymbol{v} = (1,1,1,0)$, $\boldsymbol{w} = (1,0,0,1)$, то

$$\begin{aligned} \boldsymbol{u} \cdot \boldsymbol{v} &= 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 = 1, \\ \boldsymbol{u} \cdot \boldsymbol{w} &= 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 + 1 \cdot 1 = 0, \\ \boldsymbol{v} \cdot \boldsymbol{w} &= 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 = 1. \end{aligned}$$

Следователно $\boldsymbol{u}$ и $\boldsymbol{w}$ са ортогонлни.

(*ii*) Нека $q = 2$ и нека $S = \{0100, 0101\}$. За да намерим $S^{\perp}$ означаваме $\boldsymbol{v} = (v_1, v_2, v_3, v_4) \in S^{\perp}$. Тогава

$$\begin{aligned} \boldsymbol{v} \cdot (0,1,0,0) = 0 \quad &\Rightarrow \quad v_2 = 0, \\ \boldsymbol{v} \cdot (0,1,0,1) = 0 \quad &\Rightarrow \quad v_2 + v_4 = 0. \end{aligned}$$

Оттук получаваме $v_2 = v_4 = 0$. Тъй като v_1 и v_3 са или 0 или 1, то заключаваме, че

$$S^{\perp} = \{0000, 0010, 1000, 1010\}.$$

Теорема 2.14. Нека S е подпространство на $\mathbb{F}_q^n$. Тогава имаме

$$\dim\langle S\rangle + \dim S^{\perp} = n.$$

Доказателство. Теоремата очевидно е вярна, когато $\langle S\rangle = \{\mathbf{0}\}$. Нека сега $\dim\langle S\rangle = k \geq 1$ и да изберем базис $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ на $\rangle S\langle$. Трябва да покажем, че $\dim S^{\perp} = \dim\langle S\rangle^{\perp} = n - k$. Да отбележим, че $\boldsymbol{x} \in S^{\perp}$ тогава и само тогава, когато

$$\boldsymbol{v}_1 \cdot \boldsymbol{x} = \ldots = \boldsymbol{v}_1 \cdot \boldsymbol{x} = 0,$$

което е еквивалентно с това $\boldsymbol{x}$ да удовлетворява системата $A\boldsymbol{x}^t = \mathbf{0}$, където A е $k \times n$ матрица, чийто i-ти ред е $\boldsymbol{v}_i$.

Редовете на A са линейно независими, откъдето следва, че $A\boldsymbol{x}^t = \mathbf{0}$ е хомогенна система от k линейно независими уравнения с n неизвестни. От линейната алгебра е известно, че решенията на такава система образуват векторно пространство с размерност $n - k$.

Пример 2.15. Нека $q = 2$, $n = 4$ и $S = \{0100, 0101\}$. Тогава

$$\langle S\rangle = \{0000, 0100, 0001, 0101\}.$$

Да отбележим, че $\dim\langle S\rangle = 2$. В Пример 2.13 пресметнахме, че

$$\langle S\rangle^{\perp} = \{0000, 0010, 1000, 1010\}.$$

Очевидно $\{0010, 1000\}$ е базис на $S^{\perp}$ и така $\dim S^{\perp} = 2$. С това проверихме, че

$$\dim\langle S\rangle + \dim S^{\perp} = 2 + 2 = 4 = n.$$

2.2 Линейни кодове

В този раздел ще въведем понятието линеен код и ще разгледеаме някои елементарни свойства на линейните кодове.

Дефиниция 2.16. *Линеен код* с дължина n над $\mathbb{F}_q$ наричаме всяко подпространство на $\mathbb{F}_q^n$.

Пример 2.17. (*i*) Кодът $C = \{(\lambda, \lambda, \ldots, \lambda) \mid \lambda \in \mathbb{F}_q\}$ е линеен. Той се нарича *код с повторение.*

(*ii*) ($q = 2$) $C = \{00000, 01101, 10110, 11011\}$ е линеен код.

(*iii*) ($q = 3$) $C = \{0000, 1100, 2200, 0001, 0002, 1101, 1102, 2201, 2202\}$ е линеен код.

Дефиниция 2.18. Нека C е е линеен код във $\mathbb{F}_q^n$. Ортогоналното допълнение $C^\perp$ на подпространството C наричаме *ортогонален код* на C. *Размерност* на линейния код C наричаме размерността на C като векторно пространство над $\mathbb{F}_q$.

Теорема 2.19. Нека C е линеен код с дължина n над $\mathbb{F}_q$. Тогава

(*i*) $|C| = q^{\dim C}$, т.е. $\dim C = \log_q |C|$;

(*ii*) $|C|^\perp$ е линеен код и $\dim C = \dim C^\perp = n$;

(*iii*) $(C^\perp)^\perp = C$.

Доказателство. (*i*) Виж Теорема 2.10.

(*ii*) Виж Теорема 2.14.

(*iii*) Съгласно (*ii*) имаме $\dim C^\perp + \dim (C^\perp)^\perp = n$. Така за да докажем (*iii*) е достатъчно да демонстрираме, че $C \subseteq (C^\perp)^\perp$. Нека $\boldsymbol{c} \in C$. за да проверим, че $\boldsymbol{c} \in (C^\perp)^\perp$ е достатъчно да покажем, че $\boldsymbol{c} \cdot \boldsymbol{x} = 0$ за всяко $\boldsymbol{x} \in C^\perp$. Тъй като $\boldsymbol{c} \in C$ и $\boldsymbol{x} \in C^\perp$, то от дефиницията на $C^\perp$, следва, че $\boldsymbol{c} \cdot \boldsymbol{x} = 0$. С това доказахме (*iii*). □

Един линеен код C с дължина n и размерност k над $\mathbb{F}_q$ често наричаме $[n, k]_q$-код или, ако q е ясно от контекста, просто $[n, k]$-код. Един $[n, k]_q$-код е също (n, q^k)-код. Ако е известно минималното разстояние d на C, то той се нарича $[n, k, d]_q$- или просто $[n, k, d]$-код.

Дефиниция 2.20. Нека C е линеен код. Казваме, че C е *самоортогонален*, ако $C \subseteq C^\perp$. Кода C наричаме *самодуален*, ако $C = C^\perp$.

2.3 Тегло на Хеминг

Вече дефинирахме разстояние на Хеминг $d(\boldsymbol{x}, \boldsymbol{y})$ между думите $\boldsymbol{x}$ и $\boldsymbol{y}$ като броя на компонентите, в които тези думи се различават.

Дефиниция 2.21. Нека $\boldsymbol{x}$ е дума от $\mathbb{F}_q^n$. *Тегло на Хеминг* на $\boldsymbol{x}$ ще наричаме броя на ненулевите координати на $\boldsymbol{x}$, т.е.

$$w_{\mathrm{Ham}}(\boldsymbol{x}) = d(\boldsymbol{x}, \boldsymbol{0}).$$

Теглото на Хеминг на x означаваме с $w_{\mathrm{Ham}}(\boldsymbol{x})$.

Забележка 2.22. За всеки елемент $x \in \mathbb{F}_q$ дефинираме тегло на Хеминг както следва:

$$w_{\mathrm{Ham}}(x) = d(x,0) = \begin{cases} 1 & \text{ако } x \neq 0; \\ 0 & \text{ако } x = 0. \end{cases}$$

Ако запишем $\boldsymbol{x} \in \mathbb{F}_q^n$ като $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, то тегло на Хеминг може да бъде дефинирано еквивалентно като

$$w_{\mathrm{Ham}}(\boldsymbol{x}) = \sum_{i=1}^{n} w_{\mathrm{Ham}}(x_i). \tag{2.1}$$

Лема 2.23. Ако $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$, то $d(\boldsymbol{x}, \boldsymbol{y}) = w_{\mathrm{Ham}}(\boldsymbol{x} - \boldsymbol{y})$.

Доказателство. За $x, y \in \mathbb{F}_q$, $d(x,y) = 0$ тогава и само тогава, когато $x = y$, което е вярно точно когато $x - y = 0$ или, еквивалентно, $w_{\mathrm{Ham}}(x - y) = 0$. Лемата следва сега от Забележка 2.22. □

Следствие 2.24. Нека $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$, където $q = 2^h$. Тогава $d(\boldsymbol{x}, \boldsymbol{y}) = w_{\mathrm{Ham}}(\boldsymbol{x} + \boldsymbol{y})$.

За всеки два вектора $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ и $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ в $\mathbb{F}_q^n$ дефинираме

$$\boldsymbol{x} \star \boldsymbol{y} = (x_1 y_1, x_2 y_2, \ldots, x_n y_n).$$

Лема 2.25. Ако $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_2^n$, то

$$w_{\mathrm{Ham}}(\boldsymbol{x} + \boldsymbol{y}) = w_{\mathrm{Ham}}(\boldsymbol{x}) + w_{\mathrm{Ham}}(\boldsymbol{y}) - 2w_{\mathrm{Ham}}(\boldsymbol{x} \star \boldsymbol{y}).$$

Доказателство. Поради (2.1) е достатъчно е да докажем лемата само за $x, y \in \mathbb{F}_2$. В този случай тя е очевидна. □

Лема 2.26. за всяка степен на просто число q и за всекидва вектора $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ е изпълнено

$$w_{\mathrm{Ham}}(\boldsymbol{x}) + w_{\mathrm{Ham}}(\boldsymbol{y}) \geq w_{\mathrm{Ham}}(\boldsymbol{x} + \boldsymbol{y}) \geq w_{\mathrm{Ham}}(\boldsymbol{x}) - w_{\mathrm{Ham}}(\boldsymbol{y}).$$

Дефиниция 2.27. Нека C е (не непремнно линеен) код. *Минимално тегло (на Хеминг)* за C е най-малкото ненулево тегло на кодова дума от C. Минималното тегло на C означаваме със $w_{\mathrm{Ham}}(C)$.

Теорема 2.28. Нека C е линеен код над $\mathbb{F}_q$. Тогава $d(C) = w_{\mathrm{Ham}}(C)$.

Доказателство. За всеки две думи $\boldsymbol{x}, \boldsymbol{y}$ имаме $d(\boldsymbol{x}, \boldsymbol{y}) = w_{\mathrm{Ham}}(\boldsymbol{x} - \boldsymbol{y})$. По дефиниция съществуват такива вектори $\boldsymbol{x}', \boldsymbol{y}' \in C$, за които $d(\boldsymbol{x}', \boldsymbol{y}') = d(C)$. Оттук

$$d(C) = d(\boldsymbol{x}', \boldsymbol{y}') = w_{\mathrm{Ham}}(\boldsymbol{x}' - \boldsymbol{y}') \geq w_{\mathrm{Ham}}(C),$$

тъй като $\boldsymbol{x}' - \boldsymbol{y}' \in C$.

Обратно, съществува дума $\boldsymbol{z} \in C \setminus \{\boldsymbol{0}\}$, за която $w_{\mathrm{Ham}}(C) = w_{\mathrm{Ham}}(\boldsymbol{z})$, откъдето

$$w_{\mathrm{Ham}}(C) = w_{\mathrm{Ham}}(\boldsymbol{z}) = d(\boldsymbol{z}, \boldsymbol{0}) \geq d(C).$$

□

Предимства и недостатъци на линейните кодове

2.4 Пораждаща и проверочна матрица

По дефиниция един линеен код е векторно пространство, откъдето следва, че всичките му думи се иузразяват чрез базис натова пространство. Сега ще опишем алгоритми, чрез които получаваме базис на даден линеен код или на неговия ортогонален.

Дефиниция 2.29. Нека A е матрица над $\mathbb{F}_q$. *Елементарна операция по редове*, извършена върху A наричаме пробразуване по редове от следния вид:

(i) смяна на местата на два реда;

(ii) умнохзване на ред по ненулев скалар;

(iii) добавяне към даден ред на скаларно кратно на друг ред.

Дефиниция 2.30. Две матрици наричаме *еквивалентни по редове*, ако едната може да бъде получена от другата посредством редица от елементарни операции по редове.

По-долу са дадени някои добре известни факти от линейната алгебра.

(i) Всяка матрица M над $\mathbb{F}_q$ може да бъде приведена в *горна трапецовидна форма* или *редуцирана горнотрапецовидна форма* чрез редица от елементарни операции по редове. С други думи, всяка матрица е еквивлентна (по редове) на матрица в горна трапецовидна или редуцирана горна трапецовидна форма.

(ii) Всяка матрица има единствена редуцирана горнотрапецовидна форма по редове, но може да има повече стандартни форми по редове.

Алгоритъм 1.
Вход: Непразно подмножество S на $\mathbb{F}_q^n$.
Изход Базис за $C = \langle S \rangle$, линейния код, породен от S
Описание: Формираме матрица A, чиито редове са думите на Използваме елементарни операции по редове за намиране на горнотрапецоовидна форма по редове за A. Ненулевите редове на матрицата в горнотрапецовидна форма образуват базис за C.

Пример 2.31. нека $q = 3$. Да се намери базис за $C = \langle S \rangle$, където

$$S = \{12101, 20110, 01122, 11010\}.$$

$$A = \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 2 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 2 & 2 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 0 & 2 & 2 & 1 & 1 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 2 & 2 & 1 & 2 \end{pmatrix} \to \begin{pmatrix} 1 & 2 & 1 & 0 & 1 \\ 0 & 1 & 1 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Последната матрица е в горнотрапецовидна форма по редове. Съгласно Алгоритъм 1, $\{12101, 01122, 00001\}$ е базис за C.

Алгоритъм 2.
Вход: Непразно подмножество S на $\mathbb{F}_q^n$.
Изход: Базис за $C = \langle S \rangle$, линйния код, породен от S.
Описание: Образуваме матрица A, чиито стълбове са думите от S. Използваме елементарни операции по редове за привеждане на A в горна трапецовидна форма и наираме водещите стълбове в тази форма. Тогава оригиналните стълбове на A, отговарящи на водещите стълбове образуват базис за C.

Пример 2.32. Нека $q = 2$. Да се намери базис за $C = \langle S \rangle$, където

$$S = \{11101, 10110, 01011, 11010\}.$$

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Since columns 1, 2 and 4 of the REF are the leading columns, Algorithm 2 says that column 1, 2 and 4 form a basis for C, i.e. $\{11101, 10110, 11010\}$ is a basis for C.

Алгоритъм 3.
Вход: Непразно подмножество S на $\mathbb{F}_q^n$.
Изход: Базис за ортогоналния код $C^\perp$, където $C = \langle S \rangle$.
Описание: Образуваме матрица A, чиито редове са думите на S. Използваме елементарни операции по редове за да приведем A в редуцирана горностъпаловидна форма. Нека G е $k \times n$ матрица от всички ненулеви редове на редуцираната горнотрапецовидна форма:

$$A \to \begin{pmatrix} G \\ O \end{pmatrix}.$$

Матрицата G има k водещи стълба. разместваме стълбовете на G за да получим $G' = (I_k|X)$, където I_k е единичната матрица. Образуваме матрицата H' както следва:

$$H' = (-X^T|I_{n-k}),$$

където X^T е транспонираната на X. Прилагаме обратната пермутация върху стълбовете на H' за да получим H. Редовете на H образуват базис за $C^\perp$.

Пример 2.33. Нека $q = 3$. Да асе намери базис за $C^\perp$, ако редуцираната горнотрапецовидна форма за A е

$$G = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 2 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

Водещите стълбове за G са тези с номера 1, 4, 5, 7 и 9. Разместваме стълбовете на G в реда 1, 4, 5, 7, 9, 2, 3, 6, 8, 10 и получаваме матрицата

$$G' = (I_5|X) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Образуваме матрицата H' и разместваме стълбовете на H', използвайки обратната пермутация:

$$H' = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$H = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 1 & 1 \end{pmatrix}.$$

Съгласно Алгоритъм 3, редовете на H образуват базис за $C^\perp$.

Дефиниция 2.34. Една матрица G, чиито редове образуват базис на линейния код $G = C$ наричаме *пораждаща матрица на* C. *Проверочна матрица на* C наричаме всяка матрица H, която е пораждаща за ортогоналния код $C^\perp$.

Нека C е $[n, k]$ код. Тогава всяка пораждаща матрица на C е $k \times n$ матрица, а всяка проверочна матрица е с рамер $(n-k) \times n$. Тъй като едно векторно пространство обикновено има повече от един базис, то и пораждащите матрици са повече от една. Дори и базисът на един линеен код да е фикситран, то пермутацията на редове също води до различна пораждаща матрица.

Дефиниция 2.35. Ако една пораждаща матрица има вида $(I_k|X)$, то казваме, че тя е в *стандартна форма*. Една проверочна матрица е в *стандартна форма*, ако тя има вида $(Y|I_{n-k})$.

Лема 2.36. Нека C е $[n, k]$-линеен код над $\mathbb{F}_q$ с пораждаща матрица G. Тогава думата $\boldsymbol{v} \in \mathbb{F}_q^n$ принадлежи на $C^\perp$ точно кагато $\boldsymbol{v}$ е ортогонална на всеки ред от G, т.е. $\boldsymbol{v} \in C^\perp \Leftrightarrow \boldsymbol{v}G^T = \boldsymbol{0}$. По специално, ако е дадена $(n-k) \times n$ матрица H, то тя е проверочна матрица за C тогава и само тогава, когато редовете на H са линейно независими и $HG^T = O$.

Доказателство. Нека $\boldsymbol{r}_i$ е i-тия ред на G. Тогава имаме $\boldsymbol{r}_i \in C$ за всяко $i = 1, \ldots, k$ и всяка дума $\boldsymbol{c} \in C$ може да бъде записана във вида

$$\boldsymbol{c} = \lambda_1 \boldsymbol{r}_1 + \ldots + \lambda_k \boldsymbol{r}_k,$$

където $\lambda_1, \ldots, \lambda_k \in \mathbb{F}_q$.

Ако $\in C^\perp$, то $\boldsymbol{v} \cdot \boldsymbol{c} = 0$ за всяко $\boldsymbol{c} \in C$. По специално, векторът $\boldsymbol{v}$ е ортогонален на $\boldsymbol{r}_i$ за всяко $1 \leq i \leq k$, т.е. $\boldsymbol{v}G^T = \boldsymbol{0}$. Обратно, ако $\boldsymbol{v} \cdot \boldsymbol{r}_i = 0$ за всяко $1 = 1, \ldots, k$, то за всеки вектор $\boldsymbol{c} = \lambda_1 \boldsymbol{r}_1 + \ldots + \lambda_k \boldsymbol{r}_k \in C$,

$$\boldsymbol{v} \cdot \boldsymbol{c} = \lambda_1(\boldsymbol{v} \cdot \boldsymbol{r}_1) + \ldots + \lambda_k(\boldsymbol{v} \cdot \boldsymbol{r}_k) = 0.$$

Ще докажем последното твърдение. Нека H е проверочна матрица за C. Редовете на H са линейно зависими по дефиниция. Тъй като редовете на H са кодови думи от $C^\perp$, то от предишното твърдение следва,че $HG^T = O$.

Обратно, ако $HG^T = O$, то редовете на H, а следователно и пространството от редовете на H, се съдържат в $C^\perp$. Тъй като редовете на H са линейно независими, то пространството от редовете на H има размерност $n-k$, т.е пространството от редовете на H наистина съвпада с $C^\perp$. С други думи, H е проверочна матрица за C. □

Следващата лема е алтернативна, но еквивалентна на Лема 2.36.

Лема 2.37. Нека C е линеен $[n,k]$-код над $\mathbb{F}_q$ с проверочна матрица H. Векторът $\boldsymbol{v} \in \mathbb{F}_q^n$ принадлежи на C тогава и само тогава, когато $\boldsymbol{v}$ е ортогонален на всеки ред на H, т.е. $\boldsymbol{v} \in C \Leftrightarrow \boldsymbol{v}H^T = \boldsymbol{0}$. По специално, нека G е $k \times n$ матрица над $\mathbb{F}_q$; G е пораждаща матрица за C тогава и само тогава, когато редовете ѝ са линейно независими и $GH^T = O$.

Едно следствие от Лема 2.36 е следната теорема, свързваща минималното разстояние на линеен код C със свойствата на коя да е проверочна матрица на C.

Теорема 2.38. Нека C е линеен код и нека H е проверочна матрица за C. Тогава

(*i*) C има минимално разстояние $\geq d$ тогава и само тогава, когато кои да е $d-1$ стълба на H са линейно независими;

(*ii*) C има минимално разстояние $\leq d$ тогава и само тогава, когато H има d линейно зависими стълба.

Доказателство. Нека $\boldsymbol{v} = (v_1, \ldots, v_n) \in C$ е дума с тегло $e > 0$. Да приемем, че ненулевите компоненти са в позиции $i_1, \ldots, i_e$. Така $v_j = 0$, ако $j \notin \{i_1, \ldots, i_e\}$. Нека с $\boldsymbol{h}_i$, $i = 1, \ldots, n$ означим i-тия стълб на H.

Съгласно Леми 2.36 и 2.37, C съдържа ненулева дума $\boldsymbol{v} = (v_1, \ldots, v_n)$ с тегло e, чиито ненулеви компоненти са $v_{i_1}, \ldots, V_{i_e}$ тогава и само тогава, когато

$$\boldsymbol{0} = \boldsymbol{v}H^T = v_{i_1}\boldsymbol{h}_{i_1}^T + \ldots + v_{i_e}\boldsymbol{h}_{i_e}^T,$$

Което от своя страна е вярно тогава и само тогава, когато съществуват e стълба на H (а именно $\boldsymbol{h}_{i_1}, \ldots, \boldsymbol{h}_{i_e}$), които са линейно зависими.

Твърдението, че минималното разстояние на C е $\geq d$ е еквивалентно на това, че C не съдържа ненулева дума с теглот $\leq d-1$, което е изпълнено точно когато кои да е $\leq d-1$ стълба на H са линейно независими. Това доказва (*i*).

Аналогично, твърдението, че минималното разстояние на C е $\leq d$ е еквивалентно на твърдението, че C съдържа ненулева дума с тегло $\leq d$, което е вярно точно когато H има $\leq d$ (а следователно и d) стълба, които са линейно завсисими. Това доказва (*ii*). □

Следствие 2.39. Нека C е линеен код и нека H е проверочна матрица за C. Следните твърдения са еквивалентни:

(i) кодът C има минимално разстояние d;

(ii) кои да е $d-1$ стълба на H са линейно независими и H има d стълба, които са линейно зависими.

Теорема 2.40. Ако $G = (I_k|X)$ е пораждаща матрица в стандартна форма за линеен $[n,k]$-код C, то $H = (-X^T|I_{n-k})$ е проверочна матрица за C.

Доказателство. Очевидно равенството $HG^T = O$ се изпълнява. Разглеждайки последните $n-k$ координати е ясно, че редовете на H са линейно независими. Сега резултатът следва от. Лема 2.36. □

Пример 2.41. Да се намери пораждаща и проверочна матрица за двоичния линеен код $C = \langle S\rangle$, където $S = \{11101, 10110, 01011, 11010\}$.

От Алгоритъм 1,

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \to \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

което е рдуцирана трапецовидна форм. От Алгоритъм 3 получаваме

$$G = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{array}\right), \quad H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

Тук G е пораждаща матрица за C, а H – проверочна матрица за C. Лесно се проверява, че $GH^T = O = HG^T$.

Да отбележим, че не всеки линеен код има пораждаща матрица в стандартна форма.

Пример 2.42. Да разгледаме двоичния линеен код $C = \{000, 001, 100, 101\}$. Тъй като $\dim C = 2$, съгласно Теорема 2.10(ii) броят на базисите за C е

$$\frac{1}{2}(2^2-1)(2^2-2) = 3.$$

По-долу тези базиси са изписани:

$$\{001, 100\}, \ \{001, 101\}, \ \{100, 101\}.$$

Следователно, C има шест пораждащи матрици:

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

Да забележим, че никоя от тях не е в стандартна форма.

2.5 Еквивалентност на линейни кодове

Дефиниция 2.43. Два (n, M) кода над $\mathbb{F}_q$ са еквивалентни, ако думите на всеки от тях могат да се получат от думите на другия чрез редица от операции от следния вид:

(i) пермутация на координатни позиции;

(ii) умножение на символите в определена позиция с елемент, различен от 0;

(iii) прилагане на автоморфизъм на полето към всички координатни позиции.

Пример 2.44. (i) Двоичният код $C = \{0000, 0101, 0010, 0111\}$ е еквивалентен на $C' = \{0000, 1100, 0001, 1101$
(ii) Троичният код $C = \{000, 011, 022\}$ е еквивалентен на кода $C' = \{000, 102, 201\}$.

Теорема 2.45. Всеки линеен код C е еквивалентен на линеен код C', имащ пораждаща матрица в стандартна форма.

Пример 2.46. Нека C е двоичен линеен код с пораждаща матрица

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

Пренареждайки стълбовете в реда 1, 3, 4, 2, 5, 6, 7 получаваме матрицата

$$G' \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right)$$

Нека C' е кодът, породен от G'. Тогава C' е еквивалентен на C и има пораждаща матрица в стандартна форма.

2.6 Кодиране с линеен код

Нека C е $[n, k, d]$-код над $\mathbb{F}_q$. C има q^k кодови думи и може да бъде използван за предаване на q^k различни съобщения. Ще идентифицираме тези съобщения с k-орките от $\mathbb{F}_q^k$. Фиксираме базис $\{\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k\}$ за C. Сега всяко от съобщенията може да бъде представено във вида

$$\boldsymbol{u} = u_1\boldsymbol{r}_1 + \ldots + u_k\boldsymbol{r}_k,$$

където $u_1, \ldots, u_k \in \mathbb{F}_q$. Нека G е пораждаща матрица за C, чиито i-ти ред е векторът $\boldsymbol{r}_i$. Ако е даден вектор $\boldsymbol{u} = (u_1, \ldots, u_k) \in \mathbb{F}_q^k$, то е ясно, че

$$\boldsymbol{v} = \boldsymbol{u}G = u_1\boldsymbol{r}_1 + \ldots + u_k\boldsymbol{r}_k$$

е кодова дума в C. Обратно, всяка дума $\boldsymbol{v}$ от C може да се запише по единствен начин като $\boldsymbol{v} = \boldsymbol{u}G$. Следователно всяка дума (съобщение) $\boldsymbol{u} \in \mathbb{F}_q^k$ може да се кодира като $\boldsymbol{v} = \boldsymbol{u}G$. Процесът на представяне на елементите $\boldsymbol{u}$ на $\mathbb{F}_q^k$ като кодови думи $\boldsymbol{v} = \boldsymbol{u}G$ наричаме *кодиране*.

Пример 2.47. Нека C е двоичен $[7,4]$-код с пораждаща матрица

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

Тогава съобщението (u_1, u_2, u_3, u_4) се кодира като

$$\begin{aligned} \boldsymbol{v} = \boldsymbol{u}G &= (u_1, u_2, u_3, u_4) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \\ &= (u_1, u_2, u_3, u_4, u_1 + u_2 + u_3, u_2 + u_3 + u_4, u_1 + u_2 + u_4). \end{aligned}$$

да отбележим, че скоростта на кода е $4/7$, т.е. само 4 бита от всички 7 се използват за предаване на съобщението.

2.7 Декодиране с линеен код

2.7.1 Съседни класове

В този раздел ще разгледаме една схема, предложена от Слепян, за декодиране в най-близкия съсед. Тя използва факта, че всеки линеен код е подгрупа на адитивната група $\mathbb{F}_q^n$. По-нататъкще предполагаме, че не е запознат с теория на групите и излагаме всички необходими факти.

Дефиниция 2.48. Нека C е линеен код с дължина n над $\mathbb{F}_q$ и нека $\boldsymbol{u} \in \mathbb{F}_q^n$ е вектор с дължина n. Множеството от вектори *съседен клас* на C, определен от $\boldsymbol{u}$

$$\boldsymbol{u} + C = \{\boldsymbol{u} + \boldsymbol{v} \mid \boldsymbol{v} \in C.$$

ще наричаме *съседен клас* на C, определен от $\boldsymbol{u}$.

Тъй като $\mathbb{F}_q^n$ е абелева група по отношение на събирането, имаме $\boldsymbol{u} + C = C + \boldsymbol{u}$. Всеки линеен код с дължина n е подгрупа на $\mathbb{F}_q^n$, откъдето понятието съседен клас на линеен код съвпада с традиционното понятие за съседен клас в теория на групите.

Пример 2.49. Нека $q = 2$ и нека $C = \{000, 101, 010, 111\}$. Тогава

$$\begin{aligned} 000 + C &= \{000, 101, 010, 111\}, \\ 001 + C &= \{001, 100, 011, 110\}, \\ 010 + C &= \{010, 111, 000, 101\}, \\ 100 + C &= \{100, 001, 110, 011\}, \\ 011 + C &= \{011, 110, 001, 100\}, \\ 101 + C &= \{101, 000, 111, 010\}, \\ 110 + C &= \{110, 011, 100, 001\}, \\ 111 + C &= \{111, 010, 101, 000\}. \end{aligned}$$

да отбележим, че

$$000+C=010+C=101+C=111+C=C;\ 001+C=100+C=011+C=110+C=\mathbb{F}_2^3\setminus C.$$

Теорема 2.50. Нека C е линеен $[n,k,d]$-код над крайното поле $\mathbb{F}_q$. Тогава

(*i*) всеки вектор от $\mathbb{F}_q^n$ се съдържа в някой съседен клас на C;

(*ii*) ако $\boldsymbol{u}\in\mathbb{F}_q^n$, то $|\boldsymbol{u}+C|=|C|=q^k$;

(*iii*) за всички $\boldsymbol{u},\boldsymbol{v}\in\mathbb{F}_q^n$, от $\boldsymbol{u}\in\boldsymbol{v}+C$ следва $\boldsymbol{u}+C=\boldsymbol{v}+C$;

(*iv*) два съседни класа или съвпадат или не се присичат;

(*v*) броят на различните съседни класове на C е q^{n-k};

(*vi*) за всички $\boldsymbol{u},\boldsymbol{v}\in\mathbb{F}_q^n$, $\boldsymbol{u}-\boldsymbol{v}\in C$ тогава и само тогава, когато $\boldsymbol{u}$ и $\boldsymbol{v}$ са в един и същи съседен клас.

Доказателство. (*i*) Векторът $\boldsymbol{v}$ се съдържа в съседния клас $\boldsymbol{v}+C$.

(*ii*) По дефиниция, $\boldsymbol{u}+C$ има не повече от $|C|=q^k$ елементи. Ясно е, че два елемента $\boldsymbol{u}+\boldsymbol{c}$ и $\boldsymbol{u}+\boldsymbol{c}'$ на $\boldsymbol{u}+C$ са равни тогава и само тогава, когато $\boldsymbol{c}=\boldsymbol{c}'$. Следователно $|\boldsymbol{u}+C|=|C|=q^k$.

(*iii*) От дефиницията на $\boldsymbol{v}+C$ следва, че $\boldsymbol{u}+C\subseteq\boldsymbol{v}+C$. Тогава от (*ii*) следва, че $\boldsymbol{u}+C=\boldsymbol{v}+C$.

(*iv*) Да разгледаме съседните класове $\boldsymbol{u}+C$ и да $\boldsymbol{v}+C$ допуснем, че $\boldsymbol{x}\in(\boldsymbol{u}+C)\cap(\boldsymbol{v}+C)$. Тъй като $\boldsymbol{x}\in\boldsymbol{u}+C$, от (*iii*) получаваме, че $\boldsymbol{u}+C=\boldsymbol{x}+C$. Аналогично, $\boldsymbol{v}+C=\boldsymbol{x}+C$. Следователно $\boldsymbol{u}+C=\boldsymbol{v}+C$.

(*v*) Следва от (*i*), (*ii*) и (*iv*).

(*vi*) Ако $\boldsymbol{u}-\boldsymbol{v}=\boldsymbol{c}\in C$, то $\boldsymbol{u}=\boldsymbol{v}+\boldsymbol{c}\in\boldsymbol{v}+C$, откъдето $\boldsymbol{u}+C=\boldsymbol{v}+C$. От (*i*), $\boldsymbol{u}\in\boldsymbol{u}+C$ и $\boldsymbol{v}\in\boldsymbol{v}+C$, така че $\boldsymbol{u}$ и $\boldsymbol{v}$ са в един и същи съседен клас.

Обратно, да предположим, че $\boldsymbol{u}$ и $\boldsymbol{v}$ са в един и същи съседен клас $\boldsymbol{x}+C$. Тогава $\boldsymbol{u}=\boldsymbol{x}+\boldsymbol{c}$ и $\boldsymbol{v}=\boldsymbol{x}+\boldsymbol{c}'$, където $\boldsymbol{c},\boldsymbol{c}'\in C$. Следователно $\boldsymbol{u}-\boldsymbol{v}=\boldsymbol{c}-\boldsymbol{c}'\in C$. □

Пример 2.51. Съседните класоиве на двоичния линеен $[4,2]$-код $C=\{0000,1011,0101,1110\}$ са следните:

$0000+C$:	0000	1011	0101	1110
$1000+C$:	1000	0011	1101	0110
$0100+C$:	0100	1111	0001	1010
$0010+C$:	0010	1001	0111	1100

Горната таблица наричаме *стандартна таблица на Слепиан.*

Дефиниция 2.52. Една дума с минимално тегло в съседен клас наричаме *лидер на съседния клас.*

В горния пример, векторите от първия стълб са лидери на съответните съседни класове. Да отбележим, че съседният клас $0100+C$ има и друг лидер – вектора 0001.

2.7.2 Декодиране в най-близкия съсед

Нека C е линеен код. Нека е изпратена кодовата дума $\boldsymbol{x}$, а е получена думата $\boldsymbol{y}$, с което *векторът грешка* е равен на

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{x} \in \boldsymbol{y} + C.$$

Да забележим, че векторът-грешка и получената дума са в един и същ съседен клас. Задача на декодера е да реши въз основа на получената дума $\boldsymbol{y}$, коя е изпратената кодова дума или, еквивалентно, кой е векторът грешка.

Тъй като по вероятни са вектори-грешка са с малко тегло, то декодирането в най-близкия съсед,приложено за за линеен код, работи по следния начин. При получаване дума $\boldsymbol{y}$ избираме дума с минимално тегло $\boldsymbol{e}$ от съседния клас $\boldsymbol{y} + C$ и заключаваме, че $\boldsymbol{x}' = \boldsymbol{y} - \boldsymbol{e}$ е изпратената кодова дума.

Пример 2.53. Нека C е двоичният $[4, 2]$-код от Пример 2.51: $C = \{0000, 1011, 0101, 1110\}$. Записваме стандартната таблица за C, която беше построена и в Пример 2.51.

кодови думи $\rightarrow$	0000	1011	0101	1110
	1000	0011	1101	0110
	0100	1111	0001	1010
	0010	1001	0111	1100
	$\uparrow$			
	лидери на съседни класове			

Да допуснем, че получената дума е $\boldsymbol{y} = 1101$. Векторът $\boldsymbol{y}$ е във втория съседен клас. Лидерът на този съседен клас е 1000 (и това е единственият възможен лидер в този съседен клас). Следователно, $1101 - 1000 = 0101$ е най-вероятната изпратена дума. Това е думата, която се намира във първа позиция на стълба, съдържащ получената дума.

Сега нека допуснем, че $\boldsymbol{y} = 1111$. Векторът $\boldsymbol{y}$ е в третия съседен клас. В този съседен клас имаме две думи с минимално тегло. Това означава, че има два възможни избора за лидер на съседен клас. В примера ние избираме лидерът да бъде 0100. Ако бяхме избрали 0001 за лидер, то бихме получили малко по различна таблица. В случаите, когато съседния клас на получената дума има повече от един възможен лидер, действията, които извършваме, зависят от приетата декодираща схема. Можем да извършим т.нар. *непълно декодиране* и да поискаме повторно предаване на същата дума. Ако правим *пълно декодиране*, ние избираме за вектор-грешка по произволен начин дума с минимално тегло, да речем 0100. Така заключаваме, че изпратената кодова дума е най-вероятно $1111 - 0100 = 1011$. Ако бяхме избрали 0001 за вектор-грешка, то бихме декодирали $\boldsymbol{y}$ като $1111 - 0001 = 1110$.

2.7.3 Синдромно декодиране

Декодиращата схема, основана на стандартни таблици работи прилично, когато дължината n на използвания линеен код е малка, но за големи n може да отнеме значителен ресурс от време и памет. За да намалим използваните ресурси можем да се възползваме

от синдромно декодиране за идентифициране на съседния клас, на който принадлежи получената дума.

Дефиниция 2.54. Нека C е линеен $[n,k,d]$-код над $\mathbb{F}_q$ и нека H е проверочна матрица на C. За всяка дума $y \in \mathbb{F}_q^n$ *синдром на* $\boldsymbol{y}$ наричаме думата $S(\boldsymbol{y}) = \boldsymbol{y}H^T \in \mathbb{F}_q^{n-k}$.

От дефиницията е ясно, че синдромът на една дума зависи от избора на проверочна матрица H. Следователно е смислено да означаваме синдрома с $S_H(\boldsymbol{y})$ за да подчертаем тази зависимост. Много често матрицата H се подразбира и няма опасност от объркване. В тези случаи за упростяване на означенията ще изпускаме индекса H.

Теорема 2.55. Нека C е линеен $[n,k,d]$-код над $\mathbb{F}_q$ и нека H е проверочна матрица за C. За всеки два вектора $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{F}_q^n$ е изпълнено

(i) $S(\boldsymbol{u}+\boldsymbol{v}) = S(\boldsymbol{u}+S(\boldsymbol{v})$;

(ii) $S(\boldsymbol{u}) = \boldsymbol{0}$ тогава и само тогава, когато $\boldsymbol{u}$ е кодова дума от C;

(iii) $S(\boldsymbol{u}) = S(\boldsymbol{v})$ тогава и само тогава, когато $\boldsymbol{u}$ и $\boldsymbol{v}$ са в един и същ съседен клас по C.

Доказателство. (i) следва непосредствено от дефиницията на синдром.

(ii) По дефиниция $S(\boldsymbol{u} = \boldsymbol{0}$ тогава и само тогава, когато $\boldsymbol{u}H^t = \boldsymbol{0}$, което е еквивалентно на $\boldsymbol{u} \in C$.

(iii) Това следва от (i), (ii) и Теорема 2.50. □

От точка (iii) на Теорема 2.55 следва, че всеки съседен клас може да бъде идентифициран с неговия синдром, тъйкато всички думи имат един и същи синдром. С други думи съществува взаимно-еднозначно съответствие между съседните класове и синдромите. Последните са вектори от $\mathbb{F}_q^{n-k}$ и броят им не надхвърля q^{n-k}. Съгласно Теорема 2.50(v) съществуват q^{n-k} съседни класове, следователно броят на синдромите е точно q^{n-k} сындромес. Става ясно, че всички вектори от $\mathbb{F}_q^{n-k}$ се появяват като синдроми.

Дефиниция 2.56. Една таблица, съдържаща лидерите на съседните класове заедно съсъответните им симдроми се нарича *стандартна таблица за декодиране.*

Стандартна таблица за пълно декодиране в най-близкия съсед може да се построи по следния начин:

Стъпка 1: Написваме всички съседни класове на кода и избираме лидер от всеки съседен клас, т.е. дума $\boldsymbol{u}$ с минимално тегло в класа.

Стъпка 2: Намираме проверочна матрица H за кода и пресмятаме синдрома $S(\boldsymbol{u}) = \boldsymbol{u}H^T$ за всеки лидер $\boldsymbol{u}$.

При непълно декодиране в най-близкия съсед, ако в стъпка 1 съседен клас има повече от една дума с минимално тегло, то вмест лидер поставяме символа "$*$" в таблицата на синдромите за да покажем, че в този случай е необходимо повторно предаване на изпратената дума.

Пример 2.57. Приемаме, че ще използваме декодиране в най-близкия съсед. Конструираме таблица на синдромите за кода $C = \{0000, 1011, 0101, 1110\}$. От съседните класове, намерени по-рано избираме за лидери думите 0000, 1000, 0100 и 0010. Една проверочна матрица за използвания код е

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

По-нататък конструираме таблицата със синдромите за C. Да отбележим, че всяка дума с дължина 2 се появява точно веднъж като синдром

Лидер $\boldsymbol{u}$	Синдром $S(\boldsymbol{u})$
0000	00
1000	11
0100	01
0010	10

таблицата съсиндромите за C в случая на непълно декодиране в най-близкия съсед изглежда така.

Лидер $\boldsymbol{u}$	Синдром $S(\boldsymbol{u})$
0000	00
1000	11
$*$	01
0010	10

□

Съществуването на единствен лидер в даден съседен клас съответства на грешка, която може да бъде поправена при използване на непълно декодиране в най-близкия съсед. Лидер на съседн клас (не непременно единствен) съответства на грешка, която може да бъде поправена при използване на пълно декодиране в най-близкия съсед.

Пример 2.58. Да допуснем, че ще използваме пълно декодиране в най-близкия съсед и да построим таблицата със синдромите за двоичния линеен код, зададен с проверочната матрица H, където

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

Минималното разстояние на C е 3, тъй като H няма повтарящи се стълбове (никои два стълба не са линейно зависими), а вторият, петият и шестият стълбове са линейно зависими. Тъй като $\lfloor (d-1)/2 \rfloor = 1$, то всички вектори-грешка с тегло 0 или 1 ще бъдат лидери на съседни класове. Ще пресметнем синдрома за всеки един от тях, с което получаваме и първите седем реда на таблицата със синдромите. Тъй като всеки вектор с дължина 3 се появява като синдром, то последният лидер $\boldsymbol{u}$ трябва да има синдром $\boldsymbol{u}H^T = 101$. Ясно е също, че този лидер е с тегло поне 2. Започваме да търсим $\boldsymbol{u}$ сред наличните думи (несъдържащи се в в съседните класове на избраните лидери)

с минимално тегло, т.е. тегло 2. Така намираме три думи, които са кандидати за лидери на последния съседен клас: 000101, 001010, 110000. Тъй като сме приели пълно декодиране в най-близкия съсед, избираме произволна от тях, да речем 000101, за лидер на съседен клас и с това завършваме конструирането на таблицата със синдромите.

Лидер $\boldsymbol{u}$	Синдром $S(\boldsymbol{u})$
000000	000
100000	110
010000	011
001000	111
000100	100
000010	010
000001	001
000101	101

Ще отбележим, че ако бяхме използвали непълно декодиране в най-близкия съсед, то лидерът 000101 в последния ред щеше да бъде заместен със "$*$". □

Сега можем да използваме следната процедура за синдромно декодираме:

Стъпка 1: При получена дума $\boldsymbol{y}$, пресмятаме синдрома $S(\boldsymbol{y})$.

Стъпка 2: Намираме лидера на съседен клас $\boldsymbol{u}$, съответстващ на синдрома $S(\boldsymbol{y}) = S(\boldsymbol{u})$ в таблицата със синдромите.

Стъпка 3: Декодираме $\boldsymbol{y}$ като $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{u}$.

Пример 2.59. Нека $q = 2$ и $C = \{0000, 1011, 0101, 1110\}$. Ще използваме таблицата със синдромите, конструирана в Пример 2.57 за да декодираме (i) $\boldsymbol{y} = 1101$ и (ii) $\boldsymbol{y} = 1111$.

(i) $\boldsymbol{y} = 1101$. Синдромът е $S(\boldsymbol{y}) = \boldsymbol{y}H^T = 11$. Съответният лидер на съседен клас е 1000. Следователно, $1101 + 1000 = 0101$ е най-вероятната кодова дума.

(ii) $\boldsymbol{y} = 1111$. Синдромът е $S(\boldsymbol{y}) = \boldsymbol{y}H^T = 01$. Съответният му лидер на съседен клас е 0100. Следователно, най-вероятната кодова дума е $1111 + 0100 = 1011$. □

2.8 LDPC-кодове

LDPC-кодовете с адвоични линейни кодове, чиято проверочна матрица е разредена, т.е. има относително малко единици. Те са дефинирани още през 1963 г. Галагър [**?**], но веднага са забравени. Важността на LDPC-кодовете се дължи на бързите алгоритми за декодиране, които са особено важни за приложенията.

Всеки двоичен код C се задава с проверочна матрица $H = (h_{ij})$ като в този раздел се отказваме от изискването редовете на H да бъдат линейно независими. Така $H = (h_{ij})_{m\times n}$ е с m реда и n стълба, където $m \geq n-k$. С тази проверочна матрица свързваме двуделен граф $\Gamma(C)$, който наричаме *граф на Танър*. Графът на Танър е двуделен граф с $m+n$ върха като n от тях са свързани с координатните позиции на кодовите думи, а m са свързани с проверочните съотношения. Нека $V = \{x_1, x_2, \ldots, x_n\}$ са променливи,

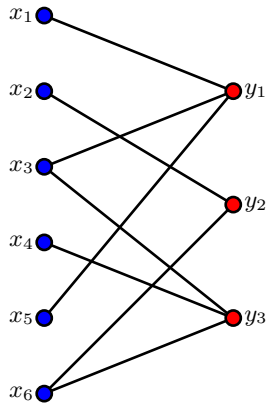
а $W = \{y_1, y_2, \ldots, y_n\}$ са проверочни съотношения. Дефинираме ребрата на този граф да са двойките $(x_i, y_j) \in V \times W$, за които $h_{ji} = 1$. Следователно проверочната матрица

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

определя еднозначно проверочните съотношения

$$\begin{aligned} y_1 &= x_1 + x_3 + x_5 \\ y_2 &= x_2 + x_6 \\ y_3 &= x_3 + x_4 + x_6 \end{aligned}$$

както и следния граф на Танър:



Очевидно $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ е в кода точно кагато всички допълнителни условия y_i са равни на нула.

Дефиниция 2.60. Един двуделен граф с множество от върхове $V \cup W$ наричаме (l, r)-*регулярен*, ако всеки връх от V е от степен l, а всеки връх от W е от степен r. Един линеен код C наричаме (l, r)-*регулярен*, ако съществува проверочна матрица H за C, която име (l, r)-регулярен граф на Танър.

Ако C е (l, r)-регулярен код с дължина n и k проверочни съотношения, то съществува проверочна матрица, имаща точно l единици във всеки стълб и точно r единици във всеки ред. Като преброим по два начина броя на единиците в тази проверочна матрица, получаваме

$$rk = nl.$$

При фиксирани l и r, но растящо n, единиците са все понарядко разположени в проверочните матрици. За да е възможно поправянето на до $\frac{d-1}{2}$ грешки, където d е минималното разстояние на кода, съответният граф трябва да е изпълнява специално свойство, дефиниция за което даваме по-долу.

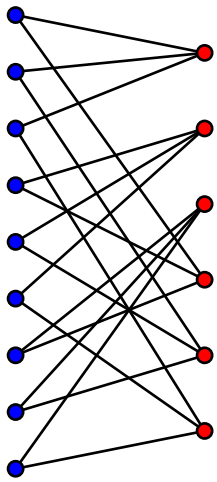
Дефиниция 2.61. Един $(l, r0$-регулярен двуделен граф с множество от върхове $V \cup W$, $|V| = n$, наричаме $(n, l, r, \alpha, \delta)$-*експандер*, $\alpha, \delta = \delta(\alpha) > 0$, ако за всяко непразно подмножество от върхове $\varnothing \neq U \subseteq V$, за което $|U| \leq \alpha|V|$ е изпълнено неравенството

$$|\partial U| > \delta|\partial U|.$$

Тук с ∂U означаваме множеството от всички върхове от W, които са свързани с поне един връх от U. Числото δ наричаме *разширяващ фактор* на експандера.

Тази дефиниция осигурява, че както и да изберем върхове от V (до някаква горна граница за броя им), то те ще са свързани с достатъчно много върхове от W.

Пример 2.62. Да разгледаме следния $(2, 3)$-регулярен двуделен граф.



Ако $\alpha = 2/9$, можем да изберем за разширяващ фактор всяко $\delta < \frac{3}{2}$. Следователно графът е $(9, 2, 3, \frac{2}{9}, \delta)$-експандер за всяко $\delta < \frac{3}{2}$. □

Всеки експандер E дефинира проверочна матрица по анчина, които описахме в началото на този раздел, а с това и двоичен линеен код $C(E)$, който ще наричаме *код на експандера*. Следващата теорема свързва параметрите на един експандер с параметрите на неговия код.

Теорема 2.63. Нека E е $(n, l, r, \alpha, \delta)$-експандер, за който $\delta \geq \frac{l}{2}$. Тогава кодът на експандер $C(E)$ има скорост $R \geq 1 - \frac{l}{r}$ и минимално разстояние $d > \alpha n$. По-специално, кодът $C(E)$ може да поправя до $\frac{\alpha n}{2}$ грешки.

Доказателство. От равенството $nl = rk$ получаваме, че кодът $C(E)$ се задава от $k = \frac{nl}{r}$ уравнения, които не са непременно линейно независими.. оттук следва, че

$$\dim C(E) \geq n - \frac{nl}{r} = n\left(1 - \frac{l}{r}\right),$$

което на свой ред води до $R = \frac{\dim C(E)}{n} = 1 - \frac{l}{r}$.

сега ще покажем, че $d > \alpha n$. Да допусмнем, че съществува ненулева дума $\boldsymbol{v} \in C(E)$, за която $w_{\mathrm{Ham}}(\boldsymbol{v}) \leq \alpha n$. Нека U е множеството на променливите, които приемат във $\boldsymbol{v}$ стойност 1. От върховете, свързани с тези променливи излизат $l|U|$ ребра. Сега съгласно дефиницията на експандер тези ребра завършват в повече от $\frac{l}{2}|U|$ върха от W, свързани с проверочни съотношения. Ясно е, че съществува връх от W, в който завършва едно ребро. Ако допусмнем, че във всеки от тези $> \frac{l}{2}|U|$ върха свършват поне две ребра, то бихме получили повече от $l|U|$ ребра, противоречие, тъй като общият брой на тези ребра е точно $l|U|$. Сега това допълнително условие приема върху $\boldsymbol{v}$ стойност 1, което означава, че $\boldsymbol{v} \notin C(E)$ в противоречие с допускането. □

Пример 2.64. Нека E е графът от Пример 2.62, който е $(9, 2, 3, \frac{2}{9}, \delta)$-експандер с $\delta < \frac{3}{2} = \frac{3}{4}l$. Съгласно Теорема 2.63 кодът $C = C(E)$ има скорост $R \geq \frac{1}{3}$, т.е. размерност ≥ 3 и минимално разстояние $d > \alpha n = 2$, т.е. $d \geq 3$. Не е трудно да се провери, че всъщност C има параметри $[9, 4, 4]$. □

Нека $C = C(E)$ е код на експандера E с параметри $(n, l, r, \alpha, \delta)$. Ще казваме, че едно проверочно съотношение е *вярно* за вектора $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$, ако то приема върху $\boldsymbol{v}$ стойност 0; в противен случай ще го наричаме *грешно.*да декодираме с помощта на следния алгоритъм:

Алгоритъм за декодиране ан LDPC-кодове

Neka $\boldsymbol{v} \in \mathbb{F}_2^n$ e poluqenata duma.

(1) Пресмятаме всички проверочни съотношения за $\boldsymbol{v}$.

(2) Намираме променлива x_i, за която броят на проверочните сътношения, в които участва, е по-голям от броя на верните. Ако такава променлива не съществува, то КРАЙ.

(3) Променяме стойността на тази променлива във $\boldsymbol{v}$.

(4) Преминаваме на стъпка (1).

Описаният алгоритъм поправя до $\frac{\alpha n}{2}$ грешки, ако разширяващият фактор $\delta = \frac{3}{4}l$.

Нека $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$ е дума, която се различава от кодова дума $\boldsymbol{c}$ в най-много $\frac{\alpha n}{2}$ позиции. Променливите, в които $\boldsymbol{v}$ се различава от $\boldsymbol{c}$ наричаме *сгрешени.* При повтарянето на редицата от стъпки (1), (2), (3) векторът $\boldsymbol{v}$ приема нова стойност чрез инвертиране на един бит, т.е. една нула се променя в единица или една единица – в нула. Ще казваме, че алгоритъмът е *в състояние* (t, s), ако $\boldsymbol{v}$ се различава от $\boldsymbol{c}$ в t позиции и s от проверочните съотношения са грешни.

Нека алгоритъмът е в състояние (t, s) със $t < \alpha n = \alpha|V|$. Нека k е броята на верните проверочни съотношения, в които влизат сгрешени променливи. От разширяващия фактор $\delta = \frac{3}{4}l$ получаваме

$$s + k > \frac{3}{4}lt. \tag{2.2}$$

Да отбелечим, че ако в едно вярно съотношение участва сгрешена променлива, то в него трябва да участва и втора сгрешена променлива; в невярно съотношение влиза

поне една сгрешена променлива. Тъй като броят на спроверочните съотношения със сгрешени променливи е tl, то

$$tl > s + 2k. \tag{2.3}$$

От (2.2) и (2.3) следва, че $tl \geq s + k + k > \frac{3}{4}lt + k$, откъдето $k < \frac{1}{4}lt$. Сега от (2.2) получаваме

$$s > \frac{lt}{2}. \tag{2.4}$$

Това означава, че съществува сгрешена променлива, за която повече от половината инцидентни с нея ребра свършват в неверни проверочни съотношения. Сега след стъпка (3) ще бъде сменена стойността на една променлива, но непременно стойността на сгрешена променлива. При това ще се намали стойността на s.

За да завърши алгоритъмът със $s = 0$, т.е. всички проверочни съотношения да са изпълнени, трябва да покажем, че при всяка итерация остава изпълнено неравенството $t < \alpha n$. Да допуснем, че $t \geq \alpha n$. Тогава от (2.4) следва, че

$$s > \frac{lt}{2} \geq \frac{l\alpha n}{2},$$

противоречие, тъй като при започване на алгоритъма имаме $s \leq \frac{l\alpha n}{2}$ и при всяка итерация s намалява стойността си.

Пример 2.65. Нека E е графът от Пример 2.62. Проверочните съотношения за кода $C(E)$ са

(1)	$x_1 + x_2 + x_3$	(4)	$x_1 + x_4 + x_7$
(2)	$x_4 + x_5 + x_6$	(5)	$x_2 + x_5 + x_8$
(3)	$x_7 + x_8 + x_9$	(6)	$x_3 + x_6 + x_9$

Да декодираме думата $\boldsymbol{v} = (0, 0, 1, 1, 1, 0, 0, 1, 1)$, в която има една грешка. Първото и четвъртото проверочни съотношения са грешни. Тъй като x_1 не участва във верни съотношения, трябав да променим $x_1 = 0$ във $x_1 = 1$. Така получаваме думата $\boldsymbol{c} = (1, 0, 1, 1, 1, 0, 0, 1, 1)$, за която всички проверочни съотношения са верни. С това думата $\boldsymbol{v}$ се декодира като $\boldsymbol{c}$. В този случай още първата итерация ни доведе до целта. □

Графи на Раманудджан.

2.9 Задачи

2.1. For each of the following sets, determine whether it is a vector space over the given field $\mathbb{F}_q$. If it is a vector space, determine the number of different bases it can have.

(a) $q = 2$, $S = \{(a, b, c, d, e) \mid a + b + c + d + e = 1\}$;

(b) $q = 3$, $T = \{(x, y, z, w) \mid xyzw = 0\}$;

(c) $q = 5$, $U = \{(\lambda + \mu, 2\mu, 3\lambda + \nu, \nu) \mid \lambda, \mu, \nu \in \mathbb{F}_5\}$;

(d) q prime, $V = \{(x_1, x_2, x_3) \mid x_1 = x_2 - x_3\}$.

2.2. For any given positive integer n and any $0 \leq k \leq n$, determine the number of distinct subspaces of $\mathbb{F}_q^n$ of dimension k.

2.3. (a) Let $\mathbb{F}_q$ be a subfield of $\mathbb{F}_r$. Show that $\mathbb{F}_r$ is a vector space over $\mathbb{F}_q$, where the vector addition and scalar multiplication are the same as the addition and multiplication of the elements in the field $\mathbb{F}_r$, respectively.

(b) Let α be a root of an irreducible polynomial of degree m over $\mathbb{F}_q$. Show that $\{1, \alpha, \ldots, \alpha^{m-1}\}$ is a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$.

2.4. Define $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \alpha + \alpha^q + \ldots + \alpha^{q^{m-1}}$ for any $\alpha \in \mathbb{F}_{q^m}$. This element is called the *trace* of α) with respect to the extension $\mathbb{F}_{q^m}/\mathbb{F}_q$.

(a) Show that $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$ is an element of $\mathbb{F}_q$ for all $\alpha \in \mathbb{F}_q^m$.

(b) Show that the map

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q} : \mathbb{F}_{q^m} \to \mathbb{F}_q,\ \alpha \to \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha)$$

is an $\mathbb{F}_q$-linear transformation, where both $\mathbb{F}_{q^m}$ and $\mathbb{F}_q$ are viewed as vector spaces over $\mathbb{F}_q$.

(c) Show that $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ is surjective.

(d) Let $\beta \in \mathbb{F}_{q^m}$. Prove that $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\beta) = 0$ if and only if there exists an element $\gamma \in \mathbb{F}_q^m$ such that $\beta = \gamma^q - \gamma$.

(e) (Transitivity of trace) Prove that

$$\mathrm{Tr}_{\mathbb{F}_{q^{rm}}/\mathbb{F}_q}(\alpha) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\mathrm{Tr}_{\mathbb{F}_{q^{rm}}/\mathbb{F}_q}(\alpha))$$

for any $\alpha \in \mathbb{F}_{q^{rm}}$

2.5. (a) Let V be a vector space over a finite field $\mathbb{F}_q$. Show that $(\lambda\boldsymbol{u}+\mu\boldsymbol{v})\cdot\boldsymbol{w} = \lambda(\boldsymbol{u}\cdot\boldsymbol{w})+\mu(\boldsymbol{v}\cdot\boldsymbol{w}$, for all $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w} \in V$ and $\lambda, \mu \in \mathbb{F}_q$.

(b) Give an example of a finite field $\mathbb{F}_q$ and a vector $\boldsymbol{u}$ defined over $\mathbb{F}_q$ with the property that $\boldsymbol{u} \neq \boldsymbol{0}$ but $\boldsymbol{u} \cdot \boldsymbol{u} = 0$.

(c) Let V be a vector space over s finite field $\mathbb{F}_q$ and let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k\}$ be a basis of V. Show that the following two statements are equivalent:

(i) $\boldsymbol{v} \cdot \boldsymbol{v}' = 0$ for all $\boldsymbol{v}, \boldsymbol{v}' \in V$,

(ii) $\boldsymbol{v}_i \cdot \boldsymbol{v}_j = 0$ for all $i, j \in \{1, \ldots, k\}$.

2.6. Let $\mathbb{F}_q$ be a finite field and let S be a subset of $\mathbb{F}_q^n$.

(a) Show that $S^\perp$ and $\langle S\rangle^\perp$ are subspaces of $\mathbb{F}_q^n$.

(b) Show that $S^\perp = \langle S\rangle^\perp$.

2.7. For each of the following sets S and corresponding finite fields $\mathbb{F}_q$, find the $\mathbb{F}_q$-linear span $\langle S\rangle$ and its orthogonal complement $S^\perp$:

(a) $S = \{101, 111, 010\}$, $q = 2$;

(b) $S = \{1020, 0201, 2001\}$, $q = 3$;

(c) $S = \{00101, 10001, 11011\}$, $q = 2$.

2.8. Determine which of the following codes are linear over $\mathbb{F}_q$:

(a) $q = 2$ and $C = \{1101, 1110, 1011, 1111\}$,

(b) $q = 3$ and $C = \{0000, 1001, 0110, 2002, 1111, 0220, 1221, 2112, 2222\}$;

(c) $q = 2$ and $C = \{00000, 11110, 01111, 10001\}$.

2.9. Let C and D be linear codes over $\mathbb{F}_q$ of the same length. Define

$$C + D = \{\boldsymbol{c} + \boldsymbol{d} \mid \boldsymbol{c} \in C, \boldsymbol{d} \in D\}.$$

Show that $C + D$ is a linear code and that $(C + D)^\perp = C^\perp \cap D^\perp$.

2.10. Determine whether each of the following statements is true or false. Justify your answer.

(a) If C and D are linear codes over $\mathbb{F}_q$ of the same length, then $C \cap D$ is also a linear code over $\mathbb{F}_q$.

(b) If C and D are linear codes over $\mathbb{F}_q$ of the same length, then $C \cup D$ is also a linear code over $\mathbb{F}_q$.

(c) If $C = \langle S \rangle$, where $S = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\} \subseteq \mathbb{F}_q^n$, then $\dim C = 3$.

(d) If $C = \langle S \rangle$, where $S = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\} \subseteq \mathbb{F}_q^n$, then

$$d(C) = \min\{w_{\mathrm{Ham}}(\boldsymbol{v}_1), w_{\mathrm{Ham}}(\boldsymbol{v}_2), w_{\mathrm{Ham}}(\boldsymbol{v}_3)\}.$$

(e) If C and D are linear codes over $\mathbb{F}_q$ with $C \subseteq D$, then $D^\perp \subseteq C^\perp$.

2.11. Determine the number of binary codes with parameters $[n, n-1, 2]$ for $n \geq 2$.

2.12. Let $\boldsymbol{u} \in \mathbb{F}_2^n$. A binary code C of length n is said to *correct the error pattern* $\boldsymbol{u}$ if and only if, for all $\boldsymbol{c}, \boldsymbol{c}' \in C$ with $\boldsymbol{c}' \neq \boldsymbol{c}$, we have $d(\boldsymbol{c}, \boldsymbol{c}+\boldsymbol{u}) < d(\boldsymbol{c}', \boldsymbol{c}+\boldsymbol{u})$. Assume that $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \mathbb{F}_2^n$ agree in at least the positions where 1 occurs in $\boldsymbol{u}_1$. Suppose that C corrects the error pattern $\boldsymbol{u}_2$. Prove that C corrects also the error pattern $\boldsymbol{u}_1$.

2.13. (a) Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_2^n$. If $\boldsymbol{x}$ and $\boldsymbol{y}$ are both of even weight, show that $\boldsymbol{x}+\boldsymbol{y}$ must have even weight.

(b) Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_2^n$. If exactly one of $\boldsymbol{x}$, $\boldsymbol{y}$ has even weight and the other has odd weight, show that $\boldsymbol{x} + \boldsymbol{y}$ must have odd weight.

(c) Prove that for a binary linear code C either all the codewords have even weight orr exactly half of the codewords have even weight.

2.14. Let C be a binary linear code with parameters $[n, k, d]$. Assume that C has at least one codeword of odd weight. Let C' denote the subset of C consisting of all the codewords of even weight. Show that C' is a binary linear code with parameters $[n, k, d']$ with $d' > d$ if d is odd, and $d' = d$ if d is even.

2.15. (a) Show that every codeword ina self-orthogonal binary code has even weight.

(b) Show that the weight of every codeword in a self-orthogonal ternary code is divisible by 3.

(c) Construct a self-orthogonal code over $\mathbb{F}_5$ such that at least one of its codewords has weight not divisible by 5.

(d) Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be codewords in a self-orthogonal binary code. Suppose the weights of $\boldsymbol{x}$ and $\boldsymbol{y}$ are both divisible by 4. Show that the weight of $\boldsymbol{x}+\boldsymbol{y}$ is also a multiple of 4.

2.16. Let C be a self-dual binary code with parameters $[n,k,d]$.

(a) Show that the all-one vector $(1,1,\ldots,1)$ is in C.

(b) Show that either all the codewords in C have weight divisible by 4, or exactly half of the codewords in C have weight divisible by 4 while the other half have even weight not divisible by 4.

(c) Let $n=6$. Determine d.

2.17. Give a parity-check matrix for a self-orthogonal binary code of length 10 and dimension 5.

2.18. Prove thet there is no self-dual binary code with parameters $[10,5,4]$.

2.19. For n odd, let C be a self-orthogonal binary $[n,(n-1)/2]$-code. Let $\mathbf{1}$ denote the all-one vector of length n and let $\mathbf{1}+C=\{\mathbf{1}+\boldsymbol{c}\mid \boldsymbol{c}\in C\}$. Show that $C^{\perp}=C\cup(\mathbf{1}+C)$.

2.20. Let C be a linear code over $\mathbb{F}_q$ of length n. For any given i with $1\le i\le n$, show that either the ith position of every codeword is 0 or every elemnt α appears in teh ith position of exactly $1/q$ codewords of C.

2.21. Let C be a linear code over $\mathbb{F}_q$ of parameters $[n,k,d]$ and suppose that, for every $1\le i\le n$, there is at least one codeword whose ith position is nonzero.

(a) Show that the sum of the weights of all the codewords in C is $n(q-1)q^{k-1}$.

(b) Show that $d\le n(q-1)q^{k-1}/(q^k-1)$.

(c) Show that there cannot be a binary linear code with parameters $[15,7,d]$, $d\ge 8$.

2.22. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two linearly independent vectors in $\mathbb{F}_q^n$ and let z denote the number of coordinates, where $\boldsymbol{x}$ and $\boldsymbol{y}$ are both zero.

(a) Show that $w_{\mathrm{Ham}}(\boldsymbol{y})+\sum_{\lambda\in\mathbb{F}_q} w_{\mathrm{Ham}}(\boldsymbol{x}+\lambda\boldsymbol{y})=q(n-z)$.

(b) Suppose further that $\boldsymbol{x}$ and $\boldsymbol{y}$ are contained in an $[n,k,d]$-code C over $\mathbb{F}_q$. Show that $w_{\mathrm{Ham}}(\boldsymbol{x})+w_{\mathrm{Ham}}(\boldsymbol{y})\le qn-(q-1)d$.

2.23. Let C be an $[n,k,d]$-code over $\mathbb{F}_q$, where $\gcd(d,q)=1$. Suppose that all the codewords of C have weight congruent to 0 or d modulo q.

(a) If $\boldsymbol{x}$ and $\boldsymbol{y}$ are linearly independent codewords such that $w_{\mathrm{Ham}}(\boldsymbol{x})\equiv w_{\mathrm{Ham}}(\boldsymbol{y})\equiv 0 \pmod q$, show that $w_{\mathrm{Ham}}(\boldsymbol{x}+\lambda\boldsymbol{y})\equiv 0 \pmod q$. for all $\lambda\in\mathbb{F}_q$.

(b) Show that $C_0=\{\boldsymbol{c}\in C\mid w_{\mathrm{Ham}}(\boldsymbol{c})\equiv 0 \pmod q\}$ is a linear subcode of C, i.e. C_0 is a linear code contained in C.

(c) Show that C cannot have a linear subcode of dimension 2 all of whose nonzero codewords have weight congruent to d modulo q. Hence deduce that C_0 has dimension $k-1$.

(d) Given a generator matrix G_0 for C_0 and a codeword $\boldsymbol{v} \in C$ of weight d, show that

$$\left(\begin{array}{c} \boldsymbol{v} \\ \hline G_0 \end{array}\right)$$

is a generator matrix for C.

2.24. Find a genmerator matrix and a parity check matrix for the linear code generated by each of the following sets, and give the parameters $[n, k, d]$ for each of these codes.

(a) $q = 2$, $S = \{1000, 0110, 0010, 0001, 1001\}$.

(b) $q = 3$, $S = \{110000, 011000, 001100, 000110, 000011\}$.

(c) $q = 2$, $S = \{10101010, 11001100, 11110000, 01100110, 00111100\}$.

2.25. Assign messages to the words in $\mathbb{F}_2^3$ as follows:

000	100	010	001	110	101	011	111
A	C	D	E	G	I	N	O

Let C be the binary code with generator matrix

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

Use G to encode the message $ENCODING$.

2.26. Find a generator matrix G' in standard form for a binary linear code equivalent to the binary linear code with the given generator matrix G:

$$\text{(a) } G = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad \text{(b) } G = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

2.27. Find a generator matrix G' in standard form for a binary linear code C' equivalent to the binary linear code C with the given parity-check matrix H:

$$\text{(a) } H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \text{(b) } H = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

2.28. Construct a binary code C of length 6 as follows: for every $(x_1, x_2, x_3) \in \mathbb{F}_2^3$, construct a 6-bit word $(x_1, x_2, \ldots, x_6) \in C$, where

$$\begin{aligned} x_4 &= x_1 + x_2 + x_3, \\ x_5 &= x_1 + x_3, \\ x_6 &= x_2 + x_3. \end{aligned}$$

(a) Show that C is a linear code.

(b) Find a generator matrix and a parity-check matrix for C.

2.29. Construct a binary code C of length 8 as follows: for every $(a, b, c, d) \in \mathbb{F}_4^2$, construct an 8-bit word $(a, b, c, d, w, x, y, z) \in C$, where

$$\begin{aligned} w &= a + b + c, \\ x &= a + b + d, \\ y &= a + c + d, \\ z &= b + c + d. \end{aligned}$$

(a) Show that C is a linear code.

(b) Find a generator matrix and a parity-check matrix for C.

(c) Show that C is exactly three-error-detecting and one-error-correcting code.

(d) Show that C is self-dual.

2.30. (a) Show that equivalent linear codes always have the same length, dimension and distance.

(b) Show that, if C and C' are equivalent, then so are their orthogonal codes $C^{\perp}$ and $(C')^{\perp}$.

2.31. Suppose that an $(n-k) \times n$ matrix H is a parity check matrix of a linear code C over $\mathbb{F}_q$. Show that, if M is an invertible $(n-k) \times (n-k)$ matrix with entries in $\mathbb{F}_q$, then MH is also a parity-check matrix for C.

2.32. Find the minimum distance of the binary linear code C with each of the following given parity-chek matrices:

$$\text{(a) } H = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{(b) } H = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

2.33. Let $n \geq 4$ and let H be a parity-check matrix for a binary linear code C of length n. Suppose that the columns of H are all distinct and that the weight of every column of H is odd. SAhow that the minimum distance of C is at least 4.

2.34. List the cosets of each of the following q-ary linear codes:

(a) $q = 3$ and $C_3 = \{000, 1010, 2020, 0101, 0202, 1111, 1212, 2121, 2222\}$,

(b) $q = 2$ and $C_2 = \{00000, 10001, 01010, 11011, 00100, 10101, 01110, 11111\}$.

2.35. Let H denote the parity-check matrix of a linear code C. Show that the coset of C whose syndrome is $\boldsymbol{v}$ contains a vector of weight t if and only if $\boldsymbol{v}$ is equal to some linear combination of t columns of H.

2.36. For m, n satisfying $2^{m-1} \leq n < 2^m$, let C be the binary $[n, n-m]$-code whose parity-check matrix H has as its ith column the binary representation of i., $1 \leq i \leq n$. Show that every coset of C contains a vector of weight ≤ 2.

2.37. Let $C \subseteq \mathbb{F}_q^n$ be a linear code with minimum distance d. Show that a word $\boldsymbol{x} \in \mathbb{F}_q^n$ is the unique coset leader of $\boldsymbol{x} + C$ if $w_{\mathrm{Ham}}(\boldsymbol{x}) \leq \lfloor (d-1)/2 \rfloor$.

2.38. Let C be a linear code of minimum distance d, where d is even. Show that some coset of C contains two vectors of weight $e+1$, where $e = \lfloor (d-1)/2 \rfloor$.

2.39. Show that

$$\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

are parity check matrices for the codes C_3 and C_2 in Problem 34, respectively. Using these parity-check matrices and assume complete decoding, construct a syndrome look-up table for each of C_3 and C_2.

2.40. Let C be the binary linear code with parity-check matrix

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

Write down a generator matrix for C and list all the codewords in C. Decode the following words:

(a) 110110; (b) 011011; (c) 101010.

2.41. Нека C е двоичен код с проверочна матрица

$$H = \left(\begin{array}{cccc|cccc|cccc|cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right).$$

(a) Докажете, че C е $[16, 6, 6]$-код.

(b) Представете графично експандера, свързан с H и определете възможно по-голям разширяващ фактор δ за $\alpha = 1/4$.

(c) Декодирайте $(1,1,0,0,1,0,1,0,0,1,1,0,0,0,0,1)$.

Глава 3

Граници в теория на кодирането

3.1 Основна задача на теория на кодирането

Даден е q-ичен (n, M, d)-код с фиксирано n. Мощността, M е мярка за ефективността на кода, а минималното разстояние d е индикация закоригиращите му способности. Разбира се, задачите за максимизиране на M и минимизиране на минималното разстояние си противоречат.

Нека C е q-ичен код с параметри (n, M, d). Да припомим, че скоростта се дефинира като $R(C) = \log_q M/n$. Сега ще въведем понятието *относително минимално разстояние*.

Дефиниция 3.1. *Относително миnмално разстояние* на q-ичния код C с параметриц (n, M, d) наричаме величината $\delta(C) = (d-1)/n$.

Относителното минимално разстояние може да бъде въведено и като d/n, но нашата дефиниция воид понякога до по-прегледни формули.

Пример 3.2. (1) Да разгледаме q-ичния код $C = \mathbb{F}_q^n$. Той има параметриц $(n, q^n, 1)$-цоде. Следователно

$$R(C) = \frac{\log_q(q^n)}{n} = 1, \;\; \delta(C) = 0.$$

Кодът има максималната възможна скорост, докато относителното минмално разстоыание е 0. Ниското относително минмално разстояние води до ниски коригиращи възможности за този код.

(2) Да разгледаме двоичния код с повторение

$$C = \{(\underbrace{0, 0, \ldots, 0}_{n}), (\underbrace{1, 1, \ldots, 1}_{n}).$$

Ясно е, че C е $(n, 2, n)$ код. Следователно

$$R(C) = \frac{\log_2(2)}{n} = \frac{1}{n} \to 0, \;\; \delta(C) = \frac{n-1}{n} \to 1,$$

когато $n \to \infty$. Този код има отлични коригиращи възможности. Но това се постига за сметката на много ниска ефективност, коеото се отразява и от неговата скорост.

(3) Да разгледаме семейството от кодовете на Хеминг. Те са кодове с параметри $(n, M, d) = (2^r - 1, 2^{n-r}, 3)$ за всички цели числа $r \geq 2$. Когато $r \to \infty$, имаме

$$R(C) = \frac{\log_2(2^{n-r})}{n} = \frac{2^r - 1 - r}{2^r - 1} \to 1, \;\; \delta(C) = \frac{2}{n} \to 0,$$

Отново забелязваме, че докато това семейство има добра асимптотична скорост, относителното минимално разстояние клони към нула, откъдето следват асимптотично лоши коригиращи способности.

Дефиниция 3.3. За дадена кодова азбука A с мощност q, $q > 1$, И фиксирани стойности за n и d, нека $A_q(n, d)$ означава максималната мощност M, за която съществува (n, M, d)-код над A. Така

$$A_q(n, d) = \max\{M \mid \text{ съществува } (n, M, d) - \text{ код } A\}.$$

Всеки (n, M, d)-код, за който мощността е максимална, т.е. $M = A_q(n, d)$, се нарича *оптимален код.*

Функцията $A_q(n, d)$ ее ценетрална за теория на кодирането. През последните години са проведни доста изследвания за определяне както на точните ѝ стойности за произволни q, n и d. Понякога задачата за определяне на стойностите на $A_q(n, d)$ се нарича *основна задача на теория на кодирането.*

За линейни кодове имаме следната дефиниция.

Дефиниция 3.4. Нека q е степен на просто число и нека n и d са цели положителни числа. С $B_q(n, d)$ означаваме максималната стойност q^k, за която съществува $[n, k, d]_q$-код над $\mathbb{F}_q$. Така

$$B_q(n, d) = \max\{q^k \mid \text{ съществува } [n, k, d]_q - \text{ ког над } \mathbb{F}_q\}.$$

В общия случай е трудно да се определят точните стойности на $A_q(n, d)$ и $B_q(n, d)$. Всепак в някои специални случаи те могат да бъдат намерени.

Теорема 3.5. Нека $q \geq 2$ е степен на просто число. Тогава

(*i*) $B_q(n, d) \leq A_q(n, d) \leq q^n$ за всяко $1 \leq d \leq n$;

(*ii*) $B_q(n, 1) = A_q(n, 1) = q^n$;

(*iii*) $B_q(n, n) = A_q(n, n) = q$;

(*iv*) $B_q(n, 2) = A_q(n, 2) = q^{n-1}$.

Доказателство.

Now we discuss the notion of the extended code. For a binary linear code, the extended code is obtained by adding a parity check coordinate. Thsi idea is generalized to codes over any finite field.

Дефиниция 3.6. For any code C over $\mathbb{F}_q$, the *extended code* of C, denoted by $\overline{C}$, is defined by

$$\overline{C} = \left\{ \left(c_1, c_2, \ldots, c_n, -\sum_{i=0}^{n} c_i \right) \mid (c_1, c_2, \ldots, c_n) \in C \right\}.$$

When $q = 2$, the extracoordinate is $\sum_{i=1}^{n} c_i$ and is called the *parity-check coordinate*.

Теорема 3.7. *If C is an (n, M, d)-code over $\mathbb{F}_q$, then $\overline{C}$ is an $(n+1, M, d')$ code over $\mathbb{F}_q$, with $d \leq d' \leq d+1$. If C is linear then so is $\overline{C}$. Moreover when C is linear,*

$$\left(\begin{array}{ccc|c} & & & 0 \\ & H & & \vdots \\ & & & 0 \\ \hline 1 & \ldots & 1 & 1 \end{array}\right)$$

is a parity-check matrix of $\overline{C}$, where H is a parity-check matrix of C.

Пример 3.8. Let us determine $A_2(5, 3)$.

Теорема 3.9. *Suppose d is odd.*

(*i*) *Then a binary (n, M, d)-code exists if and only if a binary $(n+1, M, d+1)$-code exists. Therefore, if d is odd, $A_2(n+1, d) = A_2(n, d)$.*

(*ii*) *Similarly, a binary $[n, k, d]$-linear code exists if and only if a binary $[n+1, k, d+1]$-linear code exists, so $B_2(n+1, d) = B_2(n, d)$.*

Доказателство.

This statement is equivalent to saying that if d is even, then $A_2(n, d) = A_2(n-1, d-1)$. While it is difficult to determnie the exact values of $A_q(n, d)$ and $B_q(n, d)$, several upper and lower bounds do exist. They will be discussed in the following sections.

3.2 Lower bounds

3.2.1 Sphere-covering bound

Дефиниция 3.10. Let A be an alphabet of size q, $q > 1$. For any $\boldsymbol{u} \in A^n$ and any integer $r \geq 0$, the *sphere* of radius r and center $\boldsymbol{u}$ denoted $S(\boldsymbol{u}, r)$ is the set of all vectors $\boldsymbol{v} \in A^n$ at distance at most r from $\boldsymbol{u}$:

$$S(\boldsymbol{u}, r) = \{\boldsymbol{v} \in A^n \mid d(\boldsymbol{u}, \boldsymbol{v}) \leq r\}.$$

Лема 3.11. *For all integers $r \geq 0$, a sphere of radius r in A^n, whre A is an alphabet of size $q > 1$, contains exactly $V_q^n(r)$ vectors, where*

$$V_q^n(r) = \begin{cases} \sum_{i=0}^{r} \binom{n}{i} (q-1)^i & \text{if } 0 \leq r \leq n, \\ q^n & \text{if } n \leq r. \end{cases}$$

Доказателство. Fix a vector $\boldsymbol{u} \in A^n$. We determine the number of vectors $\boldsymbol{v}$ at distance exactly i from $\boldsymbol{u}$. The number of ways in which to choose the i coordinate positions where $\boldsymbol{v}$ differs from $\boldsymbol{u}$ is $\binom{n}{i}$. For each coordinate, we have $q-1$ choices for that coordinate in $\boldsymbol{v}$. Therefore, the total number of vectors at distance exactly i from $\boldsymbol{u}$ is $\binom{n}{i}(q-1)^i$. For $0 \leq r \leq n$, the result now follows.

When $r \geq n$, we have $S(\boldsymbol{u}, r) = A^n$, hence it contains q^n vectors. □

Теорема 3.12. *For an integer $q > 1$ and integers n, d such that $1 \leq d \leq n$, we have*

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

Доказателство. (Sphere-covering bound) Let $C = \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_M\}$ be an optimal (n, M, d)-code over A, $|A| = q$. Clearly $M = A_q(n,d)$. Since C has the maximum size, there can be no word in A^n whose distance from every codeword in C is at least d. If there were such word, we could include it in C and thereby obtain an $(n, M+1, d)$-code.

Therefore, for every vector $\boldsymbol{x} \in A^n$ there is at least one word $\boldsymbol{c}_i$ in C such that $d(\boldsymbol{x}, \boldsymbol{c}_i) \leq d-1$, i.e. $\boldsymbol{x} \in S(\boldsymbol{c}_i, d-1)$. Hence every word in A^n is covered by at least one of the spheres around the codewords of C. In other words,

$$A^n \subseteq \bigcup_{i=1}^{M} S(\boldsymbol{c}_i, d-1).$$

Hence, we have

$$q^n \leq M \cdot V_q^n(d-1),$$

and the sphere-covering bound follows by Lemma 3.11. □

3.2.2 The Gilbert-Varshamov bound

the Gilbert-Varshamov bound is a lower bound for $B_q(n,d)$ known since the 1950's. There is also an asymptotic version of this bound, which we do not discuss here. For a long time the asymptotic Gilbert-Varshamov bound was the best lower bound known for to be attainable by an infinite family of linear codes, so it became a sort of benchmark for the goodness of an infinite family of linear codes. Between 1977 and 1982, V. D. Goppa constructed algebraic-geometry codes using algebraic curves over finite fields with many rational points. A major breakthrough has been achieved shortly after these discoveries, when it was shown by Tsfasman, Vladut and Zink that there are sequences of algebraic geometry codes that perform better than the asymptotic Gilbert-Varshamov bound for certain sufficiently large q.

Теорема 3.13. *(Gilbert-Varshamov bound) Let n, k, d be integers satisfying $2 \leq d \leq n$ and $1 \leq k \leq n$. If*

$$\sum_{i=0}^{d-2} \binom{n-1}{i}(q-1)^i < q^{n-k}, \tag{3.1}$$

then there exists an $[n,k]$-linear code over $\mathbb{F}_q$ with minimum distance at least d.

Доказателство. We shall show that, if (3.1) holds, then there exists an $(n-k)\times n$ matrix H over $\mathbb{F}_q$ such that every $d-1$ columns of H are linearly independent. We construct H as follows:

Let $\boldsymbol{c}_1$ be any nonzero vector in $\mathbb{F}_q^{n-k}$. Let $\boldsymbol{c}_2$ be any vector not in the span of $\boldsymbol{c}_1$. For any $2 \le j \le n$, let $\boldsymbol{c}_j$ be any linear vector that is not in the span of $d-2$ or fewer of the vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{j-1}$. Note that the number of vectors in the linear span of $d-2$ or fewer of the vectors $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_{j-1}$, $2 \le j \le n$, is given by

$$\sum_{i=0}^{d-2} \binom{j-1}{i} (q-1)^i \le \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i < q^{n-k}.$$

hence the vector $\boldsymbol{c}_j$, $2 \le j \le n$, can always be found.

The matrix H constructed in this manner is an $(n-k)\times n$ matrix, and any $d-1$ of its columns are linearly independent. The null space of H is a linear code over $\mathbb{F}_q$ of length n, of distance at least d, and of dimension at least k. By turning to a k-dimensional subspace, we obtain a linear code with the desired parameters. □

3.3 Hamming bound and perfect codes

3.3.1 Hamming bound

Теорема 3.14. *(Hamming or sphere-packing bound) For an integer $q > 1$ and integers n, d such that $1 \le d \le n$, we have*

$$A_q(n,d) \le \frac{q^n}{\sum_{i=0}^{\lfloor (d-1)/2 \rfloor} \binom{n}{i} (q-1)^i}. \tag{3.2}$$

Доказателство. let $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_M\}$ be an optimal (n, M, d)-code over A, $|A| = q$. Clearly, $M = A_q(n,d)$. Let $e = \lfloor (d-1)/2 \rfloor$. Then the packing spheres $S(\boldsymbol{c}_i, e)$ are disjoint. Hence, we have

$$\bigcup_{i=1}^{M} S(\boldsymbol{c}_i, e) \subseteq A^n,$$

where the union on the left is a disjoint union. Therefore, we have

$$M \cdot V_q^n(e) \le q^n,$$

which implies that

$$A_q(n,d) = M \le \frac{q^n}{V_q^n(e)} = \frac{q^n}{V_q^n(\lfloor (d-1)/2 \rfloor)}.$$

This completes the proof. □

Дефиниция 3.15. A code that attains the Hamming bound is called a *perfect code*.

3.3.2 Binary Hamming codes

We start by introducing the binary Hamming codes first. These codes form a special case of the more general family of q-ary Hamming codes. Binary Hamming codes are discussed separately since they are easier to describe and since they are arguably the most interesting Hamming codes.

Дефиниция 3.16. Let $r \geq 2$. A binary linear code of length $n = 2^r - 1$, with parity check-matrix H whose columns consist of all the non-zero vectors of $\mathbb{F}_2^r$, is called a *binary Hamming code* of length $2^r - 1$. It is denoted by $\mathrm{Ham}(r, 2)$.

Забележка 3.17. (1) The order of the columns of H has nit been fixed in the above definition. Hence, for each $r \geq 2$, the binary Hamming code $\mathrm{Ham}(r, 2)$ is only well-defined up to equivalence of codes.

(2) Note that the rows of H are linearly independent since H contains as columns all the r weight 1 words. Hence the rank of H is r and it is indeed a parity-check matrix.

Пример 3.18. $\mathrm{Ham}(3, 2)$: a Hamming code of length 7 is given by the following parity-check matrix:

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

Теорема 3.19. *(Properties of the binary Hamming code)*

(*i*) *All the binary Hamming codes of given length are equivalent.*

(*ii*) *The dimension of* $\mathrm{Ham}(r, 2)$ *is* $k = 2^r - 1 - r$.

(*iii*) *The minimum distance of* $\mathrm{Ham}(r, 2)$ *is* $d = 3$, *hence* $\mathrm{Ham}(r, 2)$ *is exactly single-error-correcting.*

(*iv*) *Binary Hamming codes are perfect codes.*

Доказателство. (*i*) For a given length, any parity check matrix can be obtained from another by permutation of columns. Hence, the corresponding binary Hamming codes are equivalent.

(*ii*) Since a parity-check matrix for $\mathrm{Ham}(r, 2)$ is an $r \times (2^r - 1)$ matrix, the dimension of $\mathrm{Ham}(r, 2)$ is $2^r - 1 - r$.

(*iii*) Since no two columns of H are equal, any two columns of H are linearly independent. On the other hand, H contains the columns $(100\ldots0)^T$, $(010\ldots0)^T$ and $(110\ldots,0)^T$, which are linearly dependent. Hence, by Corollary 2.39, the minimum distance of $\mathrm{Ham}(r, 2)$ is equal to 3.

(*iv*) It is easily verified that $\mathrm{Ham}(r, 2)$ satisfies the Hamming bound and is hence a perfect code. □

Since $\mathrm{Ham}(r, 2)$ is a perfect single-error-correcting code, the coset leaders are precisely the $2^r = n + 1$ vectors of length n and weight ≤ 1. Let $\boldsymbol{e}_j$ denote the vector with 1 in the j-th position and 0 elsewhere. Then the syndrome is exactly $\boldsymbol{e}_j H^T$, i.e. the transpose of the j-th column of H. Hence, if the columns of H are arranged in the order of increasing binary numbers, i.e. the j-th column is the binary representation of j, the decoding is given by:

Step 1: When $\boldsymbol{w}$ is received, calculate the syndrome $S(\boldsymbol{w}) = \boldsymbol{w}H^T$.

Step 2: If $S(\boldsymbol{w}) = \mathbf{0}$, assume $\boldsymbol{w}$ was the codeword sent.

Step 3: If $S(\boldsymbol{w}) \neq \mathbf{0}$, then $S(\boldsymbol{w}$ is the binary representation of j for some $1 \leq j \leq n$. Assuming a single error, the word $\boldsymbol{e}_j$ gives the error, so we take the sent word to be $\boldsymbol{w} - \boldsymbol{e}_j$.

Пример 3.20. Consider the Hamming code from Example 3.18. Assume the word $\boldsymbol{w} = 1001001$ has been received. The syndrome is $\boldsymbol{w}H^T = 010$. This is the binary representation of the integer 2, so the error is in the second position, i.e. the error vector is $\boldsymbol{e}_2 = 0100000$. We can then decode $\boldsymbol{w}$ as $\boldsymbol{w} - \boldsymbol{e}_2 = 1101001$.

Дефиниция 3.21. The orthogonal of the binary Hamming code $\mathrm{Ham}(r, 2)$ is called a binary *simplex code*. It is denoted by $\mathrm{Sim}(r, 2)$.

Дефиниция 3.22. The *extended binary Hamming code*, denoted by $\overline{\mathrm{Ham}(r, 2)}$ is the obtained from $\mathrm{Ham}(r, 2)$ by adding a parity-check coordinate.

Теорема 3.23. (*i*) $\overline{\mathrm{Ham}(r, 2)}$ *is a binary* $[2^r, 2^r - 1 - r, 4]$*-code.*

(*ii*) *A parity check matrix* $\overline{H}$ *for* $\overline{\mathrm{Ham}(r, 2)}$ *is*

$$\overline{H} = \left(\begin{array}{ccc|c} & & & 0 \\ & H & & \vdots \\ & & & 0 \\ \hline 1 & \dots & 1 & 1 \end{array}\right),$$

where H *is a parity-check matrix for* $\mathrm{Ham}(r, 2)$.

The transmission rate for $\overline{\mathrm{Ham}(r, 2)}$ is smaller than that of $\mathrm{Ham}(r, 2)$, but the extended code is better suited for incomplete decoding. Let $r = 3$ and take

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

Assuming that as few errors as possible have occurred, the incomplete decoding works as follows. Suppose the received vector is $\boldsymbol{w}$. Its syndrome is $S(\boldsymbol{w} = \boldsymbol{w}\overline{H}^T$. Suppose $S(\boldsymbol{w}) = (s_1, s_2, s_3, s_4)$. The $S(\boldsymbol{w})$ must fall into one of the following four categories:

(1) $s_4 = 0$ and $(s_1, s_2, s_3) = (0, 0, 0)$. In this case, $\boldsymbol{w} \in \overline{\mathrm{Ham}(r, 2)}$. We may therefore assume that there are no errors.

(2) $s_4 = 0$ and $(_1, s_2, s_3) \neq \mathbf{0}$. Since $S(\boldsymbol{w}) \neq \mathbf{0}$, at least one error must have occurred. If exactly one error occurs and it occurs in the j-th bit, then the error-vector is $\boldsymbol{e}_j$ and so $S(\boldsymbol{w} = S(\boldsymbol{e}_j$, which is the transpose of the j-th column of $\overline{H}$. An inspection of $\overline{H}$ shows that the last coordinate of every column is 1, contradicting the fact that $s_4 = 0$. Hence the assumption that exactly one error occurred is violated. We may assume that at least two errors have occurred and seek retransmission.

(3) $s_4 = 1$ and $(s_1, s_2, s_3) = \mathbf{0}$. Again, since $S(\boldsymbol{w}) \neq \mathbf{0}$, at least one error has occurred . It is easy to see that $S(\boldsymbol{w}) = S(\boldsymbol{e}_8)$, so we may assume a single error in the last coordinate.

(4) $s_4 = 1$ and $(s_1, s_2, s_3) \neq \mathbf{0}$. As before, it is easy to check that $S(\boldsymbol{w})$ must coincide with the transpose of one of the first seven columns of $\overline{H}$, say the j-th. Hence $S(\boldsymbol{w}) = S(\boldsymbol{e}_j)$, and we may assume a single error in the j-th coordinate. Given the way the columns of $\overline{H}$ are arranged, j is the number whose binary representation is (s_1, s_2, s_3).

3.3.3 q-ary Hamming codes

Let $q \geq 2$ be any prime power. Every nonzero vector $\boldsymbol{v} \in \mathbb{F}_q^r$ generates a 1-dimensional subspace of $\mathbb{F}_q^r$. Moreover, for $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{F}_q^r \setminus \{\mathbf{0}\}$, $\langle \boldsymbol{v} \rangle = \langle \boldsymbol{w} \rangle$ if and only if there is a zero scalar $\lambda \in \mathbb{F}_q \setminus \{0\}$ such that $\boldsymbol{v} = \lambda \boldsymbol{w}$. Therefore, there are exactly $(q^r - 1)/(q - 1)$ distinct subspaces of dimension 1 in $\mathbb{F}_q^r$.

Дефиниция 3.24. Let $r \geq 2$ be an integer. A q-ary linear code, whose parity-check matrix H has the property that the columns of H are made up of precisely one nonzero vector from each vector subspace of dimension 1 of $\mathbb{F}_q^r$, is called a *q-ary Hamming code*. This code is denoted by $\mathrm{Ham}(r, q)$.

For $q = 2$, the code defined above coincides with the binary Hamming code defined earlier. An easy way to write down a parity check matrix for $\mathrm{Ham}(r, q)$ is to list as columns all the nonzero r-tuples in $\mathbb{F}_q^r$ whose first nonzero entry is 1.

Теорема 3.25.

(*i*) *$Ham(r, q)$ is a $[\frac{q^r-1}{q-1}, \frac{q^r-1}{q-1} - r, 3]$-code.*

(*ii*) $\mathrm{Ham}(r, q)$ *is a perfect exactly single-error-correcting code.*

Since $\mathrm{Ham}(r, q)$ is a perfect single-error-correcting code, the coset leaders, other than $\mathbf{0}$, are exactly the vectors of weight 1. A typical coset leader is denoted by $\boldsymbol{e}_{j,b}$, $1 \leq j \leq n$, $b \in \mathbb{F}_q \setminus \{0\}$ – the vector whose jth coordinate is b and the other coordinates are all 0. Note that $S(\boldsymbol{e}_{j,b}) = b\boldsymbol{c}_j^T$, where $\boldsymbol{c}_j$ denotes the jth column of H. Decoding works as follows:

Step 1: Given a received word $\boldsymbol{w}$, calculate $S(\boldsymbol{w} = \boldsymbol{w}H^T$.

Step 2: If $S(\boldsymbol{w}) = \mathbf{0}$, then assume no errors.

Step 3: If $S(\boldsymbol{w}) \neq \mathbf{0}$, then find the unique $\boldsymbol{e}_{j,b}$ such that $S(\boldsymbol{w}) = S(\boldsymbol{e}_{j,b})$. The received word is then decoded to $\boldsymbol{w} - \boldsymbol{e}_{j,b}$.

Дефиниция 3.26. The dual of the q-ary Hamming code $\mathrm{Ham}(r, q)$ is called a *q-ary simplex code*. The q-ary simplex code is denoted by $\mathrm{Sim}(r, q)$.

3.3.4 Golay codes

3.4 Singleton bound and MDS codes

Теорема 3.27. *(Singleton bound) For any integer $q > 1$, any positive integer n, and any integer d such that $1 \leq d \leq n$, we have*

$$A_q(n, d) \leq q^{n-d+1}.$$

In particular, when q is a prime power, the parameters $[n, k, d]$ of any linear code over $\mathbb{F}_q$ satisfy

$$k + d \leq n + 1.$$

Доказателство. Consider an (n, M, d)-code C over an alphabet A of size q, where $M = A_q(n, d)$. Delete the last $d - 1$ coordinates from all codewords of C. Since the distance of C is d, after deleting the last $d - 1$ coordinates, the remaining words of length $n - d + 1$ are still distinct. The maximum number of words of length $n - d + 1$ is q^{n-d+1}, so $A_q(n, d) = M \leq q^{n-d+1}$.

The final statement of the theorem follows from the previous one by the obvious inequality $q^k \leq A_q(n, d)$. □

In the case of linear codes, we have also the following easy direct proof. Let H be the parity-check matrix for the q-ary $[n, k, d]$-code C. The rank of H is, by definition, $n - k$. Hence any $n - k + 1$ columns of H form a linerly dependent set. By Theorem 2.38, $d \leq n - k = 1$.

Дефиниция 3.28. A linear code with parameters $[n, k, d]$ such that $k + d = n + 1$ is called a *maximum distance separable code*, or *MDS code*.

An alternative way to state the Singleton bound is the following: for any q-ary code C, we have

$$R(C) + \delta(C) \leq 1.$$

Here the relative minimum distance is defined to be $\delta = d/n$. A code (not necessarily linear) is MDS if $R(C) + \delta(C) = 1$.

Теорема 3.29. *Let C be a linear code over $\mathbb{F}_q$ with parameters $[n, k, d]$. Let G and H a generator and and parity-check matrix, respectively, for C. Then, the following statements are equivalent:*

(*i*) *C is an MDS code.*

(*ii*) *Every set of $n - k$ columns of H is linearly independent.*

(*iii*) *Every set of k columns of G is linearly independent.*

(*iv*) *$C^\perp$ is an MDS code.*

Доказателство. The equivalence of (i) and (ii) follows directly from Corollary 2.39 with $d = n-k+1$. Since G is a parity-check matrix for $C^\perp$, (iii) and (iv) are also equivalent by Corollary 2.39.

$(i) \Rightarrow (iv)$ H is a generator matrix for $C^\perp$ which is of length n and dimension $n - k$. Hence we need to show that the minimum distance of $C^\perp$ is $d' = k + 1$. Suppose $d' \leq k$. Then there is a a word $\boldsymbol{c} \in C^\perp$ with at most k nonzero entries and hence at least $n - k$ zero coordinates. Permuting

the coordinates does not change the weight, so we may assume that the last $n-k$ coordinates of $\boldsymbol{c}$ are 0.

Write H as $H = (A|H')$, where A is some $(n-k)\times k$ matrix and H is a square $(n-k)\times(n-k)$ matrix. Since the columns of H' are linearly independent (for (i) and (ii) are equivalent), H' is invertible. Hence the rows of H' are linearly independent. The only way to obtain 0 in all the last $n-k$ coordinates is to use the 0-linear combination of the rows of H'. Therefore, the entire word $\boldsymbol{c}$ is the all-zero word $\mathbf{0}$. Consequently, $d' \geq k+1$. Together with the Singleton bound, it now follows that $d' = k+1$.

Since $(C^{\perp})^{\perp} = C$, the above also shows that (iv) implies (i). This completes the proof of the theorem. □

An MDS code C over $\mathbb{F}_q$ is called *trivial* if and only if C satisfies one of the following:

(i) $C = \mathbb{F}_q^n$.

(ii) C is equivalent to the code generated by $\mathbf{1} = (1, 1, \ldots, 1)$.

(iii) C is equivalent to the dual of the code generated by $\mathbf{1}$.

Otherwise, C is sid to be nontrivial. When $q = 2$, the only MDS codes are the trivial ones. This fact follows easily by considering the generator matrix in standard form.

3.5 Граница на Грийсмер

The Griesmer bound applies specifically to linear codes. It can be viewed as a generalization of the Singleton bound.

Let C be a linear code over $\mathbb{F}_q$ with parameters $[n, k]$ and suppose $\boldsymbol{c}$ is a codeword in C with $w_{\mathrm{Ham}}(\boldsymbol{c}) = w$. The *support* of $\boldsymbol{c}$, denoted by $\mathrm{Supp}(\boldsymbol{c})$, is the set of coordinate positions at which $\boldsymbol{c}$ is nonzero. Note that $|\mathrm{Supp}(\boldsymbol{c})| = w$.

Дефиниция 3.30. The *residual code of C with repsect to $\boldsymbol{c}$*, denoted by $\mathrm{Res}(C, \boldsymbol{c})$, is the code of length $n-w$ obtained from C by puncturing on all the coordinates of $\mathrm{Supp}(\boldsymbol{c})$.

Лема 3.31. *If C is an $[n, k, d]$-code over $\mathbb{F}_q$ and $\boldsymbol{c} \in C$ is a codeword of weight d, then* $\mathrm{Res}(C, \boldsymbol{c})$ *is an $[n-d, k-1, d']$-code where $d' \geq \lceil d/q \rceil$.*

Доказателство. Without loss of generality, we may replace the code C by an equivalent code so that $\boldsymbol{c} = (1, \ldots, 1, 0, \ldots, 0)$, where the first d coordinates are 1 and the other coordinates are all 0.

Note that $\mathrm{Res}(C, \boldsymbol{c})$ has dimension at most $k-1$. To see this, observe that $\mathrm{Res}(C, \boldsymbol{c})$ is a linear code. For every $\boldsymbol{x} \in \mathbb{F}_q^n$, denote by $\boldsymbol{x}'$ the vector obtained from $\boldsymbol{x}$ by deleting the first d coordinates, i.e. by puncturing on the coordinates of $\mathrm{Supp}(\boldsymbol{c})$. It is easy to see that the map $C \to \mathrm{Res}(C, \boldsymbol{c})$ given by $\boldsymbol{x} \to \boldsymbol{x}'$ is a well-defined surjective linear transformation of vector spaces whose kernel contains $\boldsymbol{c}$ and is hence a subspace of dimension at least 1. Therefore, $\mathrm{Res}(C, \boldsymbol{c})$ has dimension at most $k-1$.

We shall show that $\mathrm{Res}(C, \boldsymbol{c})$ has dimension exactly $k-1$. Suppose that the dimension is strictly less than $k-1$. Then there is a nonzero codeword $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ in C that is not a multiple of $\boldsymbol{c}$ and that has the property that $v_{d+1} = \ldots = v_n = 0$. Then $\boldsymbol{v} - v_1\boldsymbol{c}$ is a nonzero codeword that

belongs to C and thathas weight strictly less than d, a contradiction to the definition of d. Henece $\mathrm{Res}(C, \boldsymbol{c})$ has dimension $k-1$.

To show that $d' \geq \lceil d/q \rceil$, let $(x_{d+1}, \ldots, x_n)$ be any nonzero codewrod of $\mathrm{Res}(C, \boldsymbol{c})$, and let $\boldsymbol{x} = (x_1, \ldots, x_d, x_{d+1}, \ldots, x_n)$ be a corresponding word in C. By the pigeonhole principle, there is an $\alpha \in \mathbb{F}_q$ such that at least d/q coordinates of $(x_1, \ldots, x_d)$ are equal to α. Hence,

$$d \leq w_{\mathrm{Ham}}(\boldsymbol{x} - \alpha \boldsymbol{c}) \leq d - \frac{d}{q} + w_{\mathrm{Ham}}((x_{d+1}, \ldots, x_n)).$$

The inequality $d' \geq \lceil d/q \rceil$ now follows. □

Теорема 3.32. *(Griesmer bound) Let C be a q-ary code with parameters $[n, k, d]$, where $k \geq 1$. Then*

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

Доказателство. We prove the Griesmer bound by induction on k. Clearly, the theorem is true when $k = 1$.

When $k > 1$ and $\boldsymbol{c} \in C$ is a code of minimum weight d, then Lemma 3.31 shows that $\mathrm{Res}(C, \boldsymbol{c})$ is an $[n-d, k-1, d']$-code, where $d' \geq \lceil d/q \rceil$. By the inductive hypothesis, we may assume that The Griesmer bound holds for $\mathrm{Res}(C, \boldsymbol{c})$, hence

$$n - d \geq \sum_{i=0}^{k-2} \left\lceil \frac{d'}{q^i} \right\rceil \geq \left\lceil \frac{d}{q^{i+1}} \right\rceil.$$

This proves the theorem. □

Пример 3.33. The q-ary simplex code $\mathrm{Sim}(r, q)$ has parameters $[\frac{q^r-1}{q-1}, r, q^{r-1}]$, so it meets the Griesmer bound.

3.6 Plotkin bound

In this section, we discuss the Plotkin bound. This bound holds for codes for which d is large relative to n. It often gives tighter bounds than many of the other upper bounds, though it is only applicable to a relatively smalls renge of values of d.

Теорема 3.34. *let $q > 1$ be an integer and suppose that n and d satisfy $rn < d$, where $r = 1 - q^{-1}$. Then*

$$A_q(n, d) \leq \left\lfloor \frac{d}{d - rn} \right\rfloor.$$

Доказателство. Let C be an (n, M, d)-code over an alphabet A of size q. Set

$$T = \sum_{\boldsymbol{c} \in C} \sum_{\boldsymbol{c}' \in C} d(\boldsymbol{c}, \boldsymbol{c}').$$

Since $d \leq d(\boldsymbol{c}, \boldsymbol{c}')$ for $\boldsymbol{c}, \boldsymbol{c}' \in C$, $\boldsymbol{c} \neq \boldsymbol{c}'$, it follows that

$$T \geq M(M-1)d.$$

Now let $\mathcal{A}$ be an $M \times n$ array whose rows are the M codewords of C. For $i = 1, \ldots, n$, let $n_{i,a}$ denote the number of entries in the ith column of $\mathcal{A}$ that are equal to a. Clearly $\sum_{a \in A} n_{i,a} = M$ for every $i = 1, \ldots, n$. Now if we write $\boldsymbol{c} = (c_1, \ldots, c_n)$ and $\boldsymbol{c}' = (c'_1, \ldots, c'_n)$, we have

$$\begin{aligned} T &= \sum_{\boldsymbol{c} \in C} \sum_{\boldsymbol{c}' \in C} d(\boldsymbol{c}, \boldsymbol{c}') \\ &= \sum_{\boldsymbol{c} \in C} \sum_{\boldsymbol{c}' \in C} \sum_{i=1}^{n} d(c_i, c'_i) \\ &= \sum_{i=1}^{n} \left(\sum_{\boldsymbol{c} \in C} \sum_{\boldsymbol{c}' \in C} d(c_i, c'_i) \right) \\ &= \sum_{i=1}^{n} \sum_{a \in A} n_{i,A}(M - n_{i,a}) \\ &= M^2 n - \sum_{i=1}^{n} \sum_{a \in A} n_{i,a}^2 \\ &\leq M^2 n - \sum_{i=1}^{n} q^{-1} \left(\sum_{a \in A} n_{i,a} \right)^2 \\ &= M^2 r n. \end{aligned}$$

Here, we used the QM-AM inequality. □

For two-letter alphabets, i.e. $q = 2$, we can derive the following more refined version of this bound.

Теорема 3.35.

(*i*) *When d is even,*

$$A_2(n, d) \leq \begin{cases} 2\lfloor d/(2d - n) \rfloor & \text{for } n < 2d, \\ 4d & \text{for } n = 2d. \end{cases}$$

(*ii*) *When d is odd,*

$$A_2(n, d) \leq \begin{cases} 2\lfloor (d+1)/(2d + 1 - n) \rfloor & \text{for } n < 2d + 1, \\ 4d + 4 & \text{for } n = 2d + 1. \end{cases}$$

Пример 3.36. We shall illustarte that Theorem 3.35 gives a more refined bound than Theorem 3.34. Theorem 3.34 gives

$$A_2(8, 5) \leq 5, A_2(8, 6) \leq 3, A_2(12, 7) \leq 7, A_2(11, 8) \leq 3,$$

whereas Theorem 3.35 gives

$$A_2(8, 5) \leq 4, A_2(8, 6) \leq 2, A_2(12, 7) \leq 4, A_2(11, 8) \leq 2.$$

The existence of a Hadamard matrix H_n implies the existence of binary nonlinear codes with the following parameters:

$$\begin{array}{rl} (n, 2\lfloor d/(2d-n)\rfloor, d) & \text{for } d \text{ even and } d \le n < 2d; \\ (2d, 4d, d) & \text{for } d \text{ even}; \\ (n, 2\lfloor (d+1)/(2d+1-n)\rfloor, d) & \text{for } d \text{ odd and } d \le n < 2d+1; \\ (2d+1, 4d+4, d) & \text{for } d \text{ odd}. \end{array}$$

3.7 Linear programming bound

3.8 Задачи

1. Find the size, the minimum distance, the information rate, and the relative minimum distance for each of the following codes:
   (a) the binary code of all the words of weight 3;
   (b) the ternary code consisting of all the words of length 4 whose second and fourth coordinates are 0.
   (c) the code over the alphabet $\mathbb{F}_p$, p prime, consisting of all the words of length 3 whose first coordinate is $p-1$ and whose second coordinate is 1;
   (d) the repetition code over the alphabet $\mathbb{F}_p$, p prime, consisting of the following words of length n:
   $$(0,0,\ldots,0),\ (1,1,\ldots,1),\ldots,(p-1,p-1,\ldots,p-1).$$
2. For n odd, let C be a self-orthogonal binary $[n,(n-1)/2]$-code. Show that $\overline{C^{\perp}}$ is a self-dual code.
3. For any code C over $\mathbb{F}_q$ and any $\epsilon \in \mathbb{F}_q^*$, let
   $$\overline{C_\epsilon} = \left\{ \left(c_1, \ldots, c_n, \epsilon \sum_{i=1}^{n} c_i \right) \mid (c_1, \ldots, c_n) \in C \right\}.$$
   (a) If C is an (n, M, d)-code, show that $\overline{C_\epsilon}$ is an $(n+1, M, d')$-code, where $d \le d' \le d+1$.
   (b) If C is linear, show that $\overline{C_\epsilon}$ is also linear. Find a parity-check matrix of $\overline{C_\epsilon}$ in terms of a parity-check matrix of C.
4. Without using any of the bounds discussed in this chapter, show that:
   (a) $A_2(6,5) = 2$, (b) $A_2(7,5) = 2$.
5. Find an optimal binary code with $n = 3$ and $d = 2$.
6. Prove that $A_q(n,d) \le qA_q(n-1,d)$.
7. For each of the following spheres in $A^n = \mathbb{F}_2^n$, list its elements and compute its volume:
   (a) $S(110,3)$; (b) $S(1100),4)$; (c) $S(10101,2)$.

8. For each n with $4 \leq n \leq 12$, compute the Hamming bound and the sphere-covering bound for $A_2(n, 4)$.

9. Prove that a $(6, 20, 4)$-code over $\mathbb{F}_7$ cannot be an optimal code.

10. Let $q \geq 2$ and $n \geq 2$ be any integers. Show that $A_q(n, 2) = q^{n-1}$.

11. Let C be an $[n, k, d]$-code over $\mathbb{F}_q$, where $\gcd(d, q) = 1$. Suppose that all codewords of C have weight congruent to 0 or d modulo q. Show that there exists an $[n + 1, k, d + 1]$-code over $\mathbb{F}_q$.

12. Let C be an optimal code over $\mathbb{F}_{11}$ of length 12 and minimum distance 2. Show that C must have a transmission rate at least $5/6$.

13. (Gilbert-Varshamov bound for nonlinear codes) For positive integers n, M, d and $q > 1$, $1 \leq d \leq n$, show that if $(M - 1)\sum_{i=0}^{d-1} \binom{n}{i}(q - 1)^i < q^n$, then there exists a q-ary (n, M)-code of minimum distance at least d.

14. Determine whether each of the following codes exists. Justify your answer.

    (a) A binary code with parameters $(8, 29, 3)$.

    (b) A binary linear code with parameters $(8, 8, 5)$.

    (c) A binary linear code with parameters $(8, 5, 5)$.

    (d) A binary linear code with parameters $(24, 2^{12}, 8)$.

    (e) A perfect binary linear code with parameters $(63, 2^{57}, 3)$.

15. Write down a parity check matrix H for a binary Hamming code of length 15, where the jth column of H is the binary representation of j. The use H o construct the syndrome look-up table and use it to decode the following words:

    (a) 01010 01010 01000;

    (b) 11100 01110 00111;

    (c) 11001 11001 11000.

16. (a) Show that there exist no binary linear codes with parameters $[2^m, 2^m - m, 3]$ foe any $m \geq 2$.

    (b) Let C be a binary linear code with parameters $2^m, k, 4]$, for some $m \geq 2$. Show that $k \leq 2^m - m - 1$.

17. (a) Let $n \geq 3$ be an integer. Show that there is an $[n, k, 3]$-code defined over $\mathbb{F}_q$ if and only if $q^{n-k} - 1 \geq (q - 1)n$.

    (b) Find the smallest n for which there exists a ternary $[n, 5, 3]$-code.

18. (a) Let $\boldsymbol{v}$ be a nonzero vector in $\mathbb{F}_q^r$. Show that the set of vectors orthogonal to $\boldsymbol{v}$, i.e. $\{\boldsymbol{v}\}^\perp$, forms a subspace of $\mathbb{F}_q^r$ of dimension $r - 1$.

    (b) Let G be a generator matrix for the simplex code $\mathrm{Sim}(r, q)$. Show that, for a given nonzero vector $\boldsymbol{v} \in \mathbb{F}_q^r$, there are exactly $(q^{r-1} - 1)(q - 1)$-columns $\boldsymbol{c}$ of G such that $\boldsymbol{v} \cdot \boldsymbol{c} = 0$.

    (c) Show that every nonzero codeword of $\mathrm{Sim}(r, q)$ has weight q^{r-1}.

19. Determine the Hamming weight enumerators of $\mathrm{Ham}(3,2)$ and $\mathrm{Sim}(3,2)$. Verify that they satisfy the MacWilliams identity.

20. The ternary Hamming code $\mathrm{Ham}(2,3)$ is also known as the *tetracode*.

    (a) Shpw that the tetracode is a self-dual MDS code.

    (b) Without writing down all the elements of $\mathrm{Ham}(2,3)$, determine the weights of all its codewords.

    (c) Determine the Hamming weight enumerator of $\mathrm{Ham}(2,3)$ and show that the MacWilliams identity holds for $C = C^{\perp} = \mathrm{Ham}(2,3)$.

21. Let $\mathcal{G}_6$ denote the $\mathbb{F}_4$-linear code with generator matrix

$$\begin{pmatrix} 1 & 0 & 0 & 1 & \alpha & \alpha \\ 0 & 1 & 0 & \alpha & 1 & \alpha \\ 0 & 0 & 1 & \alpha & \alpha & 1 \end{pmatrix},$$

    where $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, $\alpha^2 = \alpha + 1$. This codes is known also as the *hexacode*.

    (a) Show that $\mathcal{G}_6$ is a $[6,3,4]$-code over $\mathbb{F}_4$. (Hence $\mathcal{G}_6$ is a quaternary MDS code.)

    (b) Let $\mathcal{G}_6'$ be the code obtained from $\mathcal{G}_6$ by deleting the last coordinate from every codeword. Show that $\mathcal{G}_6'$ is a Hamming code over $\mathbb{F}_4$.

22. (a) Show that the all-one vector $(1, 1, \ldots, 1)$ is in the extended Golay code G_{24}.

    (b) Deduce from (a) that G_{24} has no word of weight 20.

23. (a) Show that every word of weight 4 in $\mathbb{F}_2^{23}$ is at distance 3 from exactly one codeword in the binary Golay code G_{23}.

    (b) Use (a) to count the number of codewords of weight 7 in G_{23}.

    (c) Use (b) to show that the extended binary Golay code G_{24} contains precisely 759 codewords of weight 8.

24. Show that the extended binary Golay code G_{24} has the following weight distribution:

Weight	0	4	8	12	16	20	24
Number of codewords	1	0	759	2576	759	0	1

25. Verify the MacWilliams identity with $C = C^{\perp} = G_{24}$.

26. Prove that the extended ternary Golay code is a $[12,6,6]$-code.

27. Show that the ternary Golay code G_{11} satisfies the Hamming bound.

28. Let C be the code over $\mathbb{F}_4 = \{0, 1\alpha, \alpha^2\}$ with generator matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & \alpha & \alpha^2 \end{pmatrix}.$$

(a) Show that C is an MDS code.

(b) Write down a generator matrix for the dual $C^{\perp}$.

(c) Show that $C^{\perp}$ is an MDS code.

29. Show that the only binary MDS codes are the trivial ones.

30. Suppose that there is a q-ary MDS code of length n and dimension k, where $k < n$.

(a) Show that there is also a q-ary MDS code of length $n-1$ and dimension k.

(b) For a given $1 \leq i \leq n$, let C_i be the subcode of C consisting of all the codewords with 0 in the ith position, and let D_i be the code obtained by deleting the ith coordinate from every codeword of C_i. Show that D_i is an MDS code.

31. For each n with $9 \leq n \leq 16$, compare the Singleton, Plotkin and Hamming upper bounds for $A_2(n, 9)$.

32. Suppose that there exists a binary linear code C with parameters $[16, 8, 6]$.

(a) Let C' be the residual code of C with respect to a codeword of weight 6. Show that C' is a binary linear code with parameters $[10, 7, d']$, where $3 \leq d' \leq 4$.

(b) Argue that $d' = 3$.

(c) Show that such a C' cannot exist.

33. A binary (n, M, d)-code C is acllled a *constant-weight binary code* if there exists an integer w such that $w_{\mathrm{Ham}}(\boldsymbol{c}) = w$ for all $\boldsymbol{c} \in C$. In this case, we say that C is a constant-weight binary $(n, M, d; w)$-code.

(a) Show that the minimum distance of a constant-weight binary code is always even.

(b) Show that a constant-weight binary (n, M, d)-code satisfies $M \leq \binom{n}{w}$.

(c) Prove that a constant-weight binary $(n, M, d; w)$-code can detect at least one error.

34. Let $A_2(n, d, w)$ be the maximum possible number M of codewords in a constant-weight binary binary $(n, M, d; w)$-code. Show that

(a) $1 \leq A_2(n, d, w) \leq \binom{n}{w}$;

(b) $A_2(n, 2, w) = \binom{n}{2}$;

(c) $A_2(n, d, w) = 1$ for $d > 2w$;

(d) $A_2(n, d, w) = A_2(n, d, n - w)$.

35. Use the Griesmer bound to find an upper bound for d for the q-ary linear codes of the following n and k:

(a) $q = 2$, $n = 10$ and $k = 3$;

(b) $q = 3$, $n = 8$ and $k = 4$;

(c) $q = 4$, $n = 10$ and $k = 5$;

(d) $q = 5$, $n = 9$ and $k = 2$.

36. For a prime power q and positive integers k and u with $k > u > 0$, the *MacDonald code* $C_{k,u}$ is a q-ary code with parameters

$$[\frac{q^k - q^u}{q-1}, k, q^{k-1} - q^{u-1}],$$

that has nonzero codewords of only two possible weights: $q^{k-1} - q^{u-1}$ and q^{k-1}. Show that the MacDonald codes attain the Griesmer bound.

37. Let C ne an $[n, k, d]$-code over $\mathbb{F}_q$ and let $\boldsymbol{c} \in C$ be a codeword of weight w, where $w < dq/(q-1)$. Show that the residual code $\mathrm{Res}(C, \boldsymbol{c})$ is an $[n - w, k - 1, d']$-code, where $d' \geq d - w + \lceil w/q \rceil$.

38. Let C be a $[q^2, 4, q^2 - q - 1]$-code over $\mathbb{F}_q$.

(a) By considering $\mathrm{Res}(C, \boldsymbol{c})$ where $w_{\mathrm{Ham}}(\boldsymbol{c}) = q^2 - t$ with $2 \leq t \leq q - 1$, or otherwise, show that the only possible weights of the codewords in C are: 0, $q^2 - q - 1$, $q^2 - q$, $q^2 - 1$, and q^2.

(b) Show the existence of a $[q^2 + 1, 4, q^2 - q]$-code over $\mathbb{F}_q$.

Глава 4

Constructions of linear codes

4.1 Propagation rules

In this section, we study several constructions of new codes based on old codes. The strategy is to build codes with larger sizes or longer lengths from codes of smaller sizes or shorter lengths. First, we folrmulate some well-known propagation rules. Further rules are stated in the problems after this chapter.

Теорема 4.1. *Suppose there is an $[n, k, d]$-linear code over $\mathbb{F}_q$. Then*

(*i*) *(lengthening a code) there exists an $[n + r, k, d]$-code over $\mathbb{F}_q$ for any $r \geq 1$;*

(*ii*) *(subcodes) there exists an $[n, k - r, d]$-code over $\mathbb{F}_q$ for any $1 \leq r \leq k - 1$;*

(*iii*) *(puncturing) there exists an $[n - r, k, d - r]$-code over $\mathbb{F}_q$ for any $1 \leq r \leq d - 1$;*

(*iv*) *there exists an $[n, k, d - r]$-code over $\mathbb{F}_q$ for any $1 \leq r \leq d - 1$;*

(*v*) *there exists an $[n - r, k - r, d]$-code over $\mathbb{F}_q$ for any $1 \leq r \geq k - 1$.*

Доказателство. Let C be an $[n, k, d]$-code over $\mathbb{F}_q$.

(*i*) It suffices to show the existence of an $[n + 1, k, d]$-code over $\mathbb{F}_q$. We add a new coordinate 0 to all codewords of C:

$$\{(u_1, \ldots, u_n, 0) \mid (u_1, \ldots, u_n) \in C\}.$$

It is clear that the new code is a linear $[n + 1, k, d]$-code.

(*ii*) Let $\boldsymbol{c}$ be a nonzero codeword of C with $w_{\mathrm{Ham}}(\boldsymbol{c}) = d$. We construct a basis of C containing $\boldsymbol{c}$: $\{\boldsymbol{c}_1 = \boldsymbol{c}, \boldsymbol{c}_2, \boldsymbol{c}_k\}$. Consider the new code $\langle \boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{k-r} \rangle$ spanned by the first $k - r$ codewords in the basis. It si obvious that new code has the parameters $[n, k - r, d]$.

(*iii*) Let $\boldsymbol{c} \in C$ be a codeword of weight d. For each codeword of C, we delete a fixed set of r positions where $\boldsymbol{c}$ has nonzero coordinates. It is easy to see that the new code is an $[n - r, k, d - r]$-code.

(*iv*) The result follows by (*i*) and (*iii*).

(v) If $k = n$, then we must have that $d = 1$. Thus the space $\mathbb{F}_q^{n-r}$ is a code with parameters $[n-r, k-r, d]$.

Now we assume that $k < n$. It suffices to show the existence of an $[n-1, k-1, d]$-code for $k \geq 2$. Let C be an $[n, k, d]$-linear code over $\mathbb{F}_q$. Assume (without loss of generality) that C has a parity check matrix of the form

$$H = (I_{n-k}|X).$$

Deleting the last column of H, we obtain an $(n-k) \times (n-1)$ matrix H_1. It is clear that the row os H are linearly independent and that any $d-1$ columns of H_1 are linearly independent. Thus the linear code with H_1 as a parity-check matrix has parameters$[n-1, k-1, d_1]$ with $d_1 \geq d$. By (iv), we have an $[n-1, k-1, d]$-linear code. □

Забележка 4.2. The above theorem produces codes with worse parameters than the old ones. Usually new codes are not made using these constructions. However, they are useful when we study codes.

Следствие 4.3. *If there is an $[n, k, d]$-code over $\mathbb{F}_q$, then for any $r \geq 0$, $0 \leq s \leq k-1$ and $0 \leq t \leq d-1$, there exists an $[n+r, k-s, d-t]$-linear code over $\mathbb{F}_q$.*

Теорема 4.4. *(Direct sum of linear codes) Let C_i be an $[n_i, k_i, d_i]$-linear code over $\mathbb{F}_q$, $i = 1, 2$. Then the direct sum of C_1 and C_2 defined by*

$$C_1 \oplus C_2 = \{(\boldsymbol{c}_1, \boldsymbol{c}_2) \mid \boldsymbol{c}_1 \in C_1, \boldsymbol{c}_2 \in C_2\}$$

is an $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]$- linear code over $\mathbb{F}_q$.

Доказателство. It is an easy check that $C_1 \oplus C_2$ is a linear code over $\mathbb{F}_q$. The length of $C_1 \oplus C_2$ is clearly $n_1 + n_2$. As the size of $C_1 \oplus C_2$ is the product of the size of C_1 and the size of C_2, we obtain

$$k = \dim(C_1 \oplus C_2) = \log_q(|C_1 \oplus C_2|) = \log_q(|C_1| \cdot |C_2|) = k_1 + k_2.$$

Assume that $d_1 \leq d_2$. Let $\boldsymbol{u} \in C_1$ with $w_{\mathrm{Ham}}(\boldsymbol{u}) = d_1$. Then $(\boldsymbol{u}, \boldsymbol{0}) \in C_1 \oplus C_2$. Hence $d(C_1 \oplus C_2) \leq w_{\mathrm{Ham}}((\boldsymbol{u}, \boldsymbol{0})) = d_1$. On the other hand, for any nonzero word codeword $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in C_1 \oplus C_2$ with $\boldsymbol{c}_1 \in C_1$ and $\boldsymbol{c}_2 \in C_2$ with $\boldsymbol{c}_1 \in C_1$, $\boldsymbol{c}_2 \in C_2$, we have either $\boldsymbol{c}_1 \neq \boldsymbol{0}$ or $\boldsymbol{c}_2 \neq \boldsymbol{0}$ (or both). Thus,

$$w_{\mathrm{Ham}}((\boldsymbol{c}_1, \boldsymbol{c}_2)) = w_{\mathrm{Ham}}(\boldsymbol{c}_1) + w_{\mathrm{Ham}}(\boldsymbol{c}_2) \geq d_1.$$

This completes the proof. □

Забележка 4.5. If G_i is a generator matrix for C_i, $i = 1, 2$, then it is easy to see that the matrix

$$\begin{pmatrix} G_1 & O_{k_1 \times n_2} \\ O_{k_2 \times n_1} & G_2 \end{pmatrix}$$

is a generator matrix of $C_1 \oplus C_2$, where $O_{s \times t}$ is the s-by-t all-zero matrix.

Пример 4.6. Let $C_1 = \{000, 110, 101, 011\}$ be a binary $[3, 2, 2]$-code and let $C_2 = \{0000, 1111\}$ be a binary $[4, 1, 4]$-code. Then

$$C_1 \oplus C_2 = \{0000000, 1100000, 1010000, 0110000, 0001111, 1101111, 1011111, 0111111\}$$

is a binary $[7, 3, 2]$-linear code.

Теорема 4.7. *($(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v})$-construction) Let C be an $[n, k_i, d_i]$-code over $\mathbb{F}_q$ for $i = 1, 2$. Then the code C defined by*

$$C = \{(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v}) \mid \boldsymbol{u} \in C_1, \boldsymbol{v} \in C_2\}$$

is a $[2n, k_1 + k_2, \min\{2d_1, d_2\}]$-linear code over $\mathbb{F}_q$.

Доказателство. It is easy to verify that C is a linear code over $\mathbb{F}_q$. The length of C is clear. It is easy to show that the map

$$C_1 \oplus C_2 \to C, \ \ (\boldsymbol{c}_1, \boldsymbol{c}_2) \to (\boldsymbol{c}_1, \boldsymbol{c}_1 + \boldsymbol{c}_2)$$

is a bijection. Thus the size of C is equal to the product of the size of C_1 and that of C_2. Thus $k = k_1 + k_2$.

For any nonzero codeword $(\boldsymbol{c}_1, \boldsymbol{c}_1 + \boldsymbol{c}_2) \in C$ with $\boldsymbol{c}_1 \in C_1$ and $\boldsymbol{c}_2 \in C_2$, we have either $\boldsymbol{c}_1 \neq \boldsymbol{0}$ or $\boldsymbol{c}_2 \neq 0$ (or both). We consider two cases.
Case (1) $\boldsymbol{c}_2 = \boldsymbol{0}$. In this case, we have $\boldsymbol{c}_1 \neq \boldsymbol{0}$. Thus

$$w_{\mathrm{Ham}}((\boldsymbol{c}_1, \boldsymbol{c}_1 + \boldsymbol{c}_2)) = w_{\mathrm{Ham}}((\boldsymbol{c}_1, \boldsymbol{c}_1)) = 2w_{\mathrm{Ham}}(\boldsymbol{c}_1) \geq 2d_1 \geq \min\{2d_1, d_2\}.$$

Casse (2) $\boldsymbol{c}_2 \neq \boldsymbol{0}$. Then

$$\begin{aligned} w_{\mathrm{Ham}}((\boldsymbol{c}_1, \boldsymbol{c}_1 + \boldsymbol{c}_2)) &= w_{\mathrm{Ham}}(\boldsymbol{c}_1) + w_{\mathrm{Ham}}(\boldsymbol{c}_1 + \boldsymbol{c}_2) \\ &\geq w_{\mathrm{Ham}}(\boldsymbol{c}_1) + w_{\mathrm{Ham}}(\boldsymbol{c}_2) - w_{\mathrm{Ham}}(\boldsymbol{c}_1) \\ &= w_{\mathrm{Ham}}(\boldsymbol{c}_2) \geq d_2 \geq \min\{2d_1, d_2\}. \end{aligned}$$

This shows that $d(C) \geq \min\{2d_1, d_2\}$. On the other hand, let $\boldsymbol{x} \in C_1$ and $\boldsymbol{y} \in C_2$ with $w_{\mathrm{Ham}}(\boldsymbol{x}) = d_1$ and $w_{\mathrm{Ham}}(\boldsymbol{y}) = d_2$. Then $(\boldsymbol{x}, \boldsymbol{x}), (\boldsymbol{0}, \boldsymbol{y}) \in C$ and

$$d(C) \leq w_{\mathrm{Ham}}((\boldsymbol{x}, \boldsymbol{x})) = 2d_1, \ \ d(C) \leq w_{\mathrm{Ham}}((\boldsymbol{0}, \boldsymbol{y})) = d_2.$$

Thus $d(C) \leq \min\{2d_1, d_2\}$. This completes the proof. □

Забележка 4.8. Let G_i be a generator matrix of C_i for $i = 1, 2$. Then it is easy to see that the matrix

$$\begin{pmatrix} G_1 & G_1 \\ O_{k_2 \times n_1} & G_2 \end{pmatrix}$$

is a generator matrix of C, from the $(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v})$-construction in Theorem 4.7.

Пример 4.9. Let $C_1 = \{000, 110, 101, 011\}$ be a binary $[3, 2, 2]$-code, and let $C_2 = \{000, 111\}$ be a binary $[3, 1, 3]$-code. Then

$$C = \{000000, 110110, 101101, 011011, 000111, 110001, 101010, 011100\}$$

is a binary $[6, 3, 3]$-code.

Следствие 4.10. *Let A be a binary $[n, k, d]$-linear code. The the code C defined by*

$$C = \{(\boldsymbol{c}, \boldsymbol{c}) \mid \boldsymbol{c} \in A\} \cup \{(\boldsymbol{c}, \boldsymbol{1} + \boldsymbol{c}) \mid \boldsymbol{c} \in A\}$$

is a binary $[2n, k+1, \min\{n, 2d\}]$-code.

Доказателство. In Theorem 4.7, take $C_1 = A$ and $C_2 = \{\mathbf{0}, \mathbf{1}\}$. □

Пример 4.11. Let $A = \{00, 01, 10, 11\}$ be a binary $[2, 2, 1]$-code. Then

$$\begin{aligned} C &= \{(\boldsymbol{c}, \boldsymbol{c}) \mid \boldsymbol{c} \in A\} \cup \{(\boldsymbol{c}, \mathbf{1} + \boldsymbol{c}) \mid \boldsymbol{c} \in A\} \\ &= \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}. \end{aligned}$$

is a binary $[4, 3, 2]$-code.

4.2 Reed-Muller codes

Reed-Muller codes are among the oldest codes and have found widespread applications.For each postiive integer m and each integer r with $0 \leq r \leq m$, there is an rth order Reed-Muller code $\mathcal{R}(r, m)$ which is binary linear code with parameters

$$[2^m, \binom{m}{0} + \binom{m}{1} + \ldots \binom{m}{r}, 2^{m-r}]$$

. The code $\mathcal{R}(1, 5)$ was used by Mariner 9 to transmit black and white pictures of Mars to Earth in 1972. Reed-Muller code admit a special decoding called the Reed Decoding. There are generalizations to nonbinary fields. In this section, we concentrate on binary Reed-Muller codes.

Дефиниция 4.12. The *first order Reed-Muller codes* $\mathcal{R}(1, m)$ are binary codes defined, for all integers $m \geq 1$, recursively as follows:

(*i*) $\mathcal{R}(1, 1) = \mathbb{F}_2^2 = \{00, 01, 10, 11\}$;

(*ii*) for $m \geq 1$

$$\mathcal{R}(1, m + 1) = \{(\boldsymbol{u}, \boldsymbol{u}) \mid \boldsymbol{u} \in \mathcal{R}(1, m)\} \cup \{(\boldsymbol{u}, \boldsymbol{u} + \mathbf{1}) \mid \boldsymbol{u} \in \mathcal{R}(1, m)\}.$$

Пример 4.13. We have

$$\mathcal{R}(1, 2) = \{0000, 0101, 1010, 1111, 0011, 0110, 1001, 1100\}.$$

A generator matrix of $\mathcal{R}(1, 2)$ IS

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Теорема 4.14. *For $m \geq 1$, the Reed-Muller code $\mathcal{R}(1, m)$ is abinary linear $[2^m, m+1, 2^{m-1}]$-code, in which every codeword except $\mathbf{0}$ and $\mathbf{1}$ has weight 2^{m-1}.*

Доказателство. It is clear that $\mathcal{R}(1, 1)$ is a binary $[2, 2, 1]$-code. Note that $\mathcal{R}(1, m)$ is obtained from $\mathcal{R}(1, m - 1)$ by the construction from Corollary 4.10. Let us assume that $\mathcal{R}(1, m - 1)$ is a binary linear code with parameters $[2^{m-1}, m, 2^{m-2}]$. Then by Corollary 4.10, $\mathcal{R}(1, m)$ is a linear code with parameters

$$[2 \cdot 2^{m-1}, m + 1, \min\{2 \cdot 2^{m-2}, 2^{m-1}\}] = [2^m, m + 1, 2^{m-1}].$$

Now we are going to prove that, except for $\mathbf{0}$ and $\mathbf{1}$, every codeword of $\mathcal{R}(1, m+1)$ has weight $2^m = 2^{(m+1)-1}$.

A word in $\mathcal{R}(1, m+1)$ is either of the type $(\boldsymbol{u}, \boldsymbol{u})$ or $(\boldsymbol{u}, \boldsymbol{u}+\mathbf{1})$, where $\boldsymbol{u}$ is a word in $\boldsymbol{u}$ is a word in $\mathcal{R}(1, m)$.

Case (1) $(\boldsymbol{u}, \boldsymbol{u})$, where $\boldsymbol{u} \in \mathcal{R}(1, m)$. The vector $\boldsymbol{u}$ is neither $\mathbf{0}$ nor $\mathbf{1}$ since otherwise $(\boldsymbol{u}, \boldsymbol{u})$ is again the zero or the all-one vector. Hence, by the induction hypothesis, $\boldsymbol{u}$ has weight 2^{m-1}. Therefore $(\boldsymbol{u}, \boldsymbol{u})$ has weight $2 \cdot 2^{m-1} = 2^m$.

Case (2) $(\boldsymbol{u}, \boldsymbol{u}+\mathbf{1})$, where $\boldsymbol{u} \in \mathcal{R}(1, m)$.

(a) If $\boldsymbol{u}$ is neither $\mathbf{0}$ nor $\mathbf{1}$, then it has weight 2^{m-1}, i.e. exactly half of the coordinates are 1. Hence, half of the coordinates of $\boldsymbol{u}+\mathbf{1}$ are 1, i.e. the weight of $\boldsymbol{u}+\mathbf{1}$ is also 2^{m-1}.Therefore, the weight of $(\boldsymbol{u}, \boldsymbol{u}+\mathbf{1})$ is exactly 2^m.

(b) If $\boldsymbol{u} = \mathbf{0}$, then $\boldsymbol{u}+\mathbf{1} = \mathbf{1}$, so again the weight of $(\mathbf{0}, \mathbf{0}+\mathbf{1})$ is 2^m.

(c) If $\boldsymbol{u} = \mathbf{1}$, then $\boldsymbol{u}+\mathbf{1} = \mathbf{0}$ and the weight of $(\mathbf{1}, \mathbf{1}+\mathbf{1})$ is 2^m.

□

Теорема 4.15.

(*i*) *A generator matrix of* $\mathcal{R}(1,1)$ *is*

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

(*ii*) *If* G_m *is a generator matrix of* $\mathcal{R}(1,m)$*, then a generator matrix of* $\mathcal{R}(1,m+1)$ *is*

$$G_m = \begin{pmatrix} G_m & G_m \\ 0\ldots0 & 1\ldots1 \end{pmatrix}.$$

Пример 4.16. Using the genertor matrix

$$G_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

we obtain teh following generator matrix for $\mathcal{R}(1,3)$:

$$G_3 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Теорема 4.17. *The orthogonal code* $\mathcal{R}(1,m)^\perp$ *is equivalent to the extended binary Hamming code* $\overline{\mathrm{Ham}(m,2)}$.

Доказателство. By Theorem 4.15, starting with

$$G_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

it is clear that G_m is of the form

$$\left(\begin{array}{c|ccc} 1 & 1 & \dots & 1 \\ \hline 0 & & & \\ \vdots & & H_m & \\ 0 & & & \end{array}\right),$$

where H_m is some matrix. Moving the first coordinate to the last and moving the first row of the matrix to the last, we obtain the following generator matrix G'_m for an equivalent code:

$$\left(\begin{array}{ccc|c} & & & 0 \\ & H_m & & \vdots \\ & & & 0 \\ \hline 1 & \dots & 1 & 1 \end{array}\right).$$

By Theorem 3.7, if we show that H_m is a parity-check matrix for $\mathrm{Ham}(m,2)$, then G'_m is a parity-check matrix for $\overline{\mathrm{Ham}(m,2)}$, so $\mathcal{R}(1,m)^\perp$ is equivalent to $\overline{\mathrm{Ham}(m,2)}$.

To show that H_m is a parity-check matrix for $\mathrm{Ham}(m,2)$, we need to show that the columns of H_m concict of all the nonzero vectors of length m. Indeed, when $m = 1, 2$, the columns of H_m consist of all the nonzero vectors of length m. Now suppose that the columns of H_m consist of all the nonzero vectors of length m, for some m. By the definition of G_m, it follows readily that the columns of H_{m+1} consist of the following:

$$\begin{pmatrix} \boldsymbol{c} \\ 0 \end{pmatrix}, \begin{pmatrix} \boldsymbol{c} \\ 1 \end{pmatrix}, \begin{pmatrix} \mathbf{0}^T \\ 1 \end{pmatrix},$$

where $\boldsymbol{c}$ is one of the columns of H_m and $\mathbf{0}$ is the zero vector of length m. It is clear that the vectors in this list make up exactly all the nonzero vectors of length $m+1$. Hence, by induction, the columns of H_m consist of all the nonzero vectors of length m. □

Finally, we give a definition for the rth order Reed-Muller codes.

Дефиниция 4.18. (*i*) The *zeroth order Reed-Muller codes* $\mathcal{R}(0,m)$, for $m \geq 0$, are defined to be the repetition codes $\{\mathbf{0}, \mathbf{1}\}$ of length 2^m.

(*ii*) The *first order Reed-Muller codes* $\mathcal{R}(1,m)$, for $m \geq 1$, are defined as in Definition 4.12.

(*iii*) For any $r \geq 2$, the *rth order Reed-Muller codes* $\mathcal{R}(r,m)$ are defined, for $m \geq r-1$, recursively by

$$\mathcal{R}(r,m+1) = \begin{cases} \mathbb{F}_2^{2^r} & \text{if } m = r-1, \\ \{(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v}) \mid \boldsymbol{u} \in \mathcal{R}(r,m), \boldsymbol{v} \in \mathcal{R}(r-1,m)\} & textifm > r-1. \end{cases}$$

4.3 Кодове над подполе

Теорема 4.19. Нека A е $[N, K, D]$-линеен код над $\mathbb{F}_{q^m}$. и нека съществува линеен $[n, m, d]$-код B над $\mathbb{F}_q$. Тогава съществува линеен код над $\mathbb{F}_q$ с параметри $[nN, mK, d']$, за който $d' = d(C) \geq dD$. По специално, съществува линеен код над $\mathbb{F}_q$ с параметри $[nN, mK, dD]$.

Доказателство. Полето $\mathbb{F}_{q^m}$ може да се разглежда като векторно пространство над $\mathbb{F}_q$. Да изберем биективна линейна трансформация ϕ между $\mathbb{F}_{q^m}$ и B. Продължаваме изображението ϕ до изображение

$$\phi^* : \mathbb{F}_{q^m}^N \to \mathbb{F}_q^{nN}, \ \ (v_1, \ldots, v_n) \mapsto (\phi(v_1), \ldots, \phi(v_n)).$$

Ясно е, че ϕ^* е линейно изображение над $\mathbb{F}_q$ от $\mathbb{F}_{q^m}^N$ във $\mathbb{F}_q^{nN}$. Изображението ϕ^* е инективно, но не и сюрективно (освен в случая, когато $n = m$).

Кодът A е $\mathbb{F}_q$-подпространство на $\mathbb{F}_{q^m}^N$. Нека C е образът на A при ϕ^*, т.е. $C = \phi^*(A)$. Тъй като ϕ^* е $\mathbb{F}_q$-линейно изображение, кодът C е подпространство на $\mathbb{F}_q^{nN}$.

Очевидно дължината на C е nN. Сега ще опредлим размерността на C. Тъй като

$$\dim_{\mathbb{F}_r} V = \log_r |V| \text{ или } |V| = r^{\dim_{\mathbb{F}_r} V},$$

имаме

$$\begin{aligned} \dim_{\mathbb{F}_q} C &= \log_q |C| \\ &= \log_q |A| (\phi^* \text{ е инективно }) \\ &= \log_q \left((q^m)^{\dim_{\mathbb{F}_{q^m}} A} \right) \\ &= \log_q q^{mK} = mK. \end{aligned}$$

СЕга ще определим минималното разстояние на C. Нека $(u_1, \ldots, u_N)$ е ненулева кодова дума от A. Ако $u_i \neq 0$ за някое $1 \leq i \leq N$, то $\phi(u_i)$ е ненулева кодова дума от B. Следователно, $w_{\mathrm{Ham}}(\phi(u_i)) \geq d$. Тъй като $(u_1, \ldots, u_N)$ има поне D ненулеви позиции, то броят на ненулевите позиции в $(\phi(u_1), \ldots, \phi(u_N))$ е поне dD. Съгласно Теорема **??**(iv) съществува и линеен $[nN, mK, dD]$-код над $\mathbb{F}_q$. □

Кода A в горната теорема наричаме *външен код*, докато кода B наричаме *вътрешен код*. Ако в Теорема 4.19 вземем за вътрешен тривиалния код, т.е. $B = \mathbb{F}_q^m$, то получаваме следния резултат.

Следствие 4.20. Ако съществува линеен код с параметри $[N, K, D]$ над $\mathbb{F}_{q^m}$, то съществува и линеен $[mN, mK, D]$-код над $\mathbb{F}_{q^m}$.

Пример 4.21. (i) Известно е, че съществува $[17, 15, 3]$-код на Хеминг над $\mathbb{F}_{16}$, както и двоичен $[8, 4, 4]$-код. От Теорема 4.19 получаваме двоичен $[136, 60, 12]$-код.

(ii) Съществува $[\frac{8^3-1}{8-1}, \frac{8^3-1}{8-1} - 3, 3] = [73, 70, 3]$-код на Хеминг над $\mathbb{F}_8$. От Следствие 4.20 получаваме линеен $[219, 210, 3]$-код. □

Пример 4.22. (i) Да разгледаме линейния код

$$A = \{(0,)), (1,\alpha), (\alpha, 1+\alpha), (1+\alpha, 1)\}$$

над $\mathbb{F}_4$, където α е корен на $1+x+x^2$. нека B е двоичният код

$$\{000, 110, 101, 011\}$$

и да разгледаме линейната трансформация от $\mathbb{F}_4$ в B, дефинирана чрез

$$\phi : 0 \mapsto 000,\ \ 1 \mapsto 110,\ \ \alpha \mapsto 101,\ \ 1+\alpha \mapsto 011.$$

Така получаваме кода

$$C := \phi^*(A) = \{000000, 110101, 101011, 011110\}.$$

Новият код C има параметри $[6, 2, 4]$.

(ii) Нека α е корен на $1+x+x^3$. Тогава $\mathbb{F}_8 = \mathbb{F}(\alpha)$. Елементите $\{1, \alpha, \alpha^2\}$ образуват базис на $\mathbb{F}_8$ над $\mathbb{F}_2$. разглеждаме изображението $\phi : \mathbb{F}_8 \to \mathbb{F}_2^3$

$$a_1 \cdot 1 + a_2 \cdot \alpha + a_3 \cdot \alpha^2 \mapsto (a_1, a_2, a_3).$$

Нека $A = \langle(\alpha, \alpha+1, a)\rangle/\mathbb{F}_8$. Съгласно Следствие 4.20 $C := \phi^*(A)$ е двоичен $[9, 3, d]$-код, където $d \geq 3$. Да напишем всички елементи на A:

$$\begin{aligned} A \ = \ & \{(0,0,0), (\alpha, \alpha+1, 1), (\alpha^2, \alpha^2+\alpha, \alpha), \\ & (\alpha+1, \alpha^2+\alpha+1, \alpha^2), (\alpha^2+\alpha, \alpha^2+1, \alpha+1), \\ & (\alpha^2+\alpha+1, 1, \alpha^2+\alpha), (\alpha^2+1, \alpha, \alpha^2+\alpha+1), (1, \alpha^2, \alpha^2+1)\}. \end{aligned}$$

Следователно,

$$\begin{aligned} C = \phi^*(A) \ = \ & \{000000000, 010110100, 001011010, 110111001, \\ & 011101110, 111100011, 101010111, 100001101\}. \end{aligned}$$

Така C е двоичен $[9, 3, 4]$-код. □

Всяко векторно пространство V над $\mathbb{F}_{q^m}$ може да се разглежда като векторно пространство над $\mathbb{F}_q$. По специално $\mathbb{F}_{q^m}^N$ е векторно пространство над $\mathbb{F}_q$ с размерност mN. Това но води до нова кодова конструкция.

Теорема 4.23. Нека C е $[N, K, D]$-код над $\mathbb{F}_{q^m}$. Кодът, дефиниран над подполето $\mathbb{F}_q$ $C|_{\mathbb{F}_q} := C \cap \mathbb{F}_q^N$ е линеен $[n, k, d]$-код над $\mathbb{F}_q$ със $n = N, k \geq mK - (m-1)N$ и $d \geq D$. По-специално, ако $mK > (m-1)N$, то съществува линеен код над $\mathbb{F}_q$ с параметри $[N, mK - (m-1)N, D]$.

Доказателство. Ясно е, че $C|_{\mathbb{F}_q}$ е линеен код, тъй катои C и $\mathbb{F}_q^N$ могат да се разглеждат като подпространства на $\mathbb{F}_{q^m}^N$ над $\mathbb{F}_q$. Дължината на $C|_{\mathbb{F}_q}$ очевидно е n. За размерността имаме

$$\begin{aligned}\dim_{\mathbb{F}_q} C|_{\mathbb{F}_q} &= \dim_{\mathbb{F}_q}(C \cap \mathbb{F}_q^N) \\ &= \dim_{\mathbb{F}_q} C + \dim_{\mathbb{F}_q} \mathbb{F}_q^N - \dim_{\mathbb{F}_q}(C + \mathbb{F}_q^N) \\ &\geq \log_q |C| + N - \dim_{\mathbb{F}_q}(\mathbb{F}_{q^m}^N) \\ &= \log_q (q^m)^K + N - \log_q q^{mN} \\ &= mK + N - mN = mK - (m-1)N.\end{aligned}$$

Тъй като $C|_{\mathbb{F}_q}$ е подмножество на C, то е ясно, че минималното тегло на Хеминг за $C|_{\mathbb{F}_q}$ е поне равно на минималното тегло на Хеминг за C, т.е. $d(C|_{\mathbb{F}_q}) \geq d(C) = D$. Прилагайки Следствие 4.3, получаваме втората част на желания резултат. □

Пример 4.24. Нека α е корен на $1 + x + x^2 \in \mathbb{F}_2[x]$. Тогава $\mathbb{F}_4 = \mathbb{F}_2[\alpha]$. Нека

$$C = \langle \{(\alpha, 0, 0), (0, \alpha + 1, 0)\} \rangle / \mathbb{F}_4.$$

Така, съгласно Теорема 4.23, $C|_{\mathbb{F}_2}$ е двоичен $[3, k, d]$-код със

$$k \geq mK - (m-1)N = 2 \cdot 2 - (2-1) \cdot 3 = 1, \;\; d \geq d(C) = 1.$$

Да изпишем елементите на C:

$$\begin{aligned}C = \{&(0,0,0), (\alpha,0,0), (1,0,0), (\alpha+1,0,0), \\ &(0,\alpha+1,0), (0,\alpha,0), (0,1,0), (\alpha,\alpha+1,0), \\ &(\alpha,\alpha,0), (\alpha,1,0), (1,\alpha+1,0), (1,\alpha,0), \\ &(1,1,0), (\alpha+1,\alpha+1,0), (\alpha+1,\alpha,0), \\ &(\alpha+1,1,0)\}.\end{aligned}$$

Ясно е, че $C|_{\mathbb{F}_2} = C \cap \mathbb{F}_2^3 = \{000, 100, 010, 110\}$. Следователно, $C|_{\mathbb{F}_2}$ е двоичен $[3, 2, 1]$-код. □

Теорема 4.25. Нека C е линеен $[N, K, D]$-код над $\mathbb{F}_{q^m}$. Тогава следата на кода C, дефиниран чрез

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C) := \{(\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_1), \ldots, \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(c_n)) \mid (c_1, \ldots, c_n) \in c\}$$

е линеен $[n, k]$-код над $\mathbb{F}_q$ с $n = N$ и $k \leq mK$.

Доказателство. Тъй като $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ е $\mathbb{F}_q$-линейна трансформация от $\mathbb{F}_{q^m}$ във $\mathbb{F}_q$, множеството $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C)$ е подпространство на $\mathbb{F}_q^n$. Ясно е, че дължината на $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C)$ е n. за размерността на този код имаме

$$\begin{aligned}\dim_{\mathbb{F}_q} \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C) &= \log_q |\,\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C)| \\ &\leq \log_q |C| = \log_q (q^m)^{\dim_{\mathbb{F}_{q^m}} C} \\ &= \log_q (q^{mK}) = mK.\end{aligned}$$

□

Пример 4.26. Да разгледаме кода $C = \{\lambda(1, \alpha, \alpha+1) \mid \lambda \in \mathbb{F}_9\}$ над $\mathbb{F}_9$, където α е корен на $2 + x + x^2 \in \mathbb{F}_3[x]$. Тогава

$$\begin{aligned} C &= \{(0,0,0), (\alpha, 1+2\alpha, 1), (1+2\alpha, 2+2\alpha, \alpha), \\ &\quad (2+2\alpha, 2, 1+2\alpha), (2, 2\alpha, 2+2\alpha), (2\alpha, 2+\alpha, 2), \\ &\quad (2+\alpha, 1+\alpha, 2\alpha), (1+\alpha, 1, 2+\alpha), (1, \alpha, 1+\alpha)\}. \end{aligned}$$

Прилагайки изображението следа $\mathrm{Tr}_{\mathbb{F}_9/\mathbb{F}_3}$, имаме

$$\begin{array}{ll} (0,0,0) \mapsto (0,0,0) & (\alpha, 1+2\alpha, 1) \mapsto (2,0,2), \\ (1+2\alpha, 2+2\alpha, \alpha) \mapsto (0,2,2) & (2+2\alpha, 2, 1+2\alpha) \mapsto (2,1,0), \\ (2, 2\alpha, 2+2\alpha) \mapsto (1,1,2) & ((2\alpha, 2+\alpha, 2) \mapsto (1,0,1), \\ (2+\alpha, 1+\alpha, 2\alpha) \mapsto (0,1,1) & (1+\alpha, 1, 2+\alpha) \mapsto (1,2,0) \\ (1, \alpha, 1+\alpha) \mapsto (2,2,1). & \end{array}$$

Следователно следата на кода C

$$\begin{aligned} \mathrm{Tr}_{\mathbb{F}_9/\mathbb{F}_3}(C) &= \{(0,0,0), (2,0,2), (0,2,2), (2,1,0), (1,1,2), \\ &\quad (1,0,1), (0,1,1), (1,2,0), (2,2,1) \end{aligned}$$

е линеен $[3,2,2]$-код над $\mathbb{F}_3$. □

Всъщност следата на код е подкод над подполе. Това се вижда от следния резултат.

Теорема 4.27. (Делсарт) За линеен код C над $\mathbb{F}_{q*m}$ е в сила

$$(C|_{\mathbb{F}_q})^\perp = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^\perp).$$

Доказателство. За да докажем, че $(C|_{\mathbb{F}_q})^\perp \supseteq \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^\perp)$ трябва да демонстрираме, че

$$\boldsymbol{c} \cdot \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\boldsymbol{a}) = 0, \text{ за всички } \boldsymbol{a} \in C^\perp \text{ и } \boldsymbol{c} \in C|_{\mathbb{F}_q}.$$

Ако $\boldsymbol{c} = (c_1, \ldots, c_n)$ и $\boldsymbol{a} = (a_1, \ldots, a_n)$, то

$$\begin{aligned} \boldsymbol{c} \cdot \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\boldsymbol{a}) &= \sum_{i=1}^{n} c_i \,\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(a_i) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}\left(\sum_{i=1}^{n} c_i a_i\right) \\ &= \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\boldsymbol{c} \cdot \boldsymbol{a}) = 0. \end{aligned}$$

Тук използваме $\mathbb{F}_q$-линейността на следата, както и факта, че $\boldsymbol{c} \cdot \boldsymbol{a} = 0$.

Сега ще покажем, че $(C|_{\mathbb{F}_q})^\perp \subseteq \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^\perp)$. Това твърдение е еквивалентно на

$$\left(\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^\perp)\right)^\perp \subseteq C|_{\mathbb{F}_q}.$$

Да предположим, че горното включване не е изпълнено. Тогава съществува някакво

$$\boldsymbol{u} \in \left(\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^\perp)\right)^\perp \setminus C|_{\mathbb{F}_q}$$

и $\boldsymbol{v} \in C^{\perp}$ със $\boldsymbol{u} \cdot \boldsymbol{v} \neq 0$. Тъй като $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}$ не е нулевото изображение, съществува елемент на $\gamma \in \mathbb{F}_{q^m}$ такъв, че $\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\gamma(\boldsymbol{u} \cdot \boldsymbol{v})) \neq 0$. Следователно

$$\boldsymbol{u} \cdot \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\gamma \boldsymbol{v}) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\boldsymbol{u} \cdot \gamma \boldsymbol{v}) = \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\gamma(\boldsymbol{u} \cdot \boldsymbol{v}) \neq 0.$$

От друга страна имаме $\boldsymbol{u} \cdot \mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\gamma \boldsymbol{v}) = 0$, защото $\boldsymbol{u} \in \left(\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(C^{\perp})\right)^{\perp}$ и $\gamma \boldsymbol{v} \in C^{\perp}$. Полученото противоречие доказва желания резултат. □

Горната теорема показва, че кодове, получени като следа, могат да бъдат получени и като подкодове над подполе.

Пример 4.28. Да разгледаме както в Пример 4.26 кода $C = \{\lambda(1, \alpha, \alpha + 1) \mid \lambda \in \mathbb{F}_9\}$ над $\mathbb{F}_9$, където α е корен на $2 + x + x^2 \in \mathbb{F}_3[x]$. Тогава съгласно Теорема 4.27 и Пример 4.26 имаме

$$\begin{aligned} C^{\perp}|_{\mathbb{F}_3} &= (\mathrm{Tr}_{\mathbb{F}_9/\mathbb{F}_3}(C))^{\perp} \\ &= \{(0,0,0),(2,0,2),(0,2,2),(2,1,0),(1,1,2), \\ &\quad (1,0,1),(0,1,1),(1,2,0),(2,2,1)\}^{\perp} \\ &= \{(0,0,0),(1,1,2),(2,2,1)\}. \end{aligned}$$

□

4.4 Problems

1. (a) Given an $[n, k, d]$-linear code over $\mathbb{F}_q$, can one always construct an $[n + 1, k + 1, d]$-code? Justify your answer.
   (b) Given an $[n, k, d]$-linear code over $\mathbb{F}_q$, can one always construct an $[n + 1, k, d + 1]$-code? Justify your answer.

2. Let C be a q-ary $[n, k, d]$-code. For a fixed $1 \leq i \leq n$, form the subset A of C consisting of the codewords with the ith position equal to 0. Delete the ith position from all the words in A to form a code D. Show that D is q-ary $[n - 1, k', d']$-linear code with

   $$k - 1 \leq k' \leq k, \;\; d' \geq d.$$

   This way of obtaining a new code is called *shortening*.

3. Suppose that

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

   is a generator matrix of a binary code C. Find find a generator matrix of A with respect to $i = 2$ using the construction in previous Exercise.

4. Let H_i be a parity-check matrix of C_i for $i = 1, 2$.
   (a) Find a parity-check matrix of $C_1 \oplus C_2$ and justify your answer.

(b) Find a parity-check matrix of the code obtained from the $(\boldsymbol{u}, \boldsymbol{u}+\boldsymbol{v})$-construction and justify your answer.

5. (a) Let $A = \{0000, 1100, 0011, 1111\}$ be a binary code Find the code C constructed from A using Corollary 4.10.

   (b) Let H be a parity-check matrix of A in (a). Find a parity-check matrix of C constructed from A using Corollary 4.10.

6. Assume that q is odd. Let C_i be an $[n, k_i, d_i]$-code over $\mathbb{F}_q$ for $i = 1, 2$. Define

$$C_1()C_2 = \{\boldsymbol{c}_1 + \boldsymbol{c}_2, \boldsymbol{c}_1 - \boldsymbol{c}_2 \mid \boldsymbol{c}_1 \in C_1, \boldsymbol{c}_2 \in C_2\}.$$

   (a) Show that $C_1()C_2$ is a $[2n, k_1 + k_2]$-linear code over $\mathbb{F}_q$.

   (b) If G_i is a generator matrix for C_i, for $i = 1, 2$, find a generator matrix for $C_1()C_2$ in terms of G_1 and G_2.

   (c) Let d be the minimum distance of $C_1()C_2$. Show that $d = 2d_2$ if $2d_2 \leq d_1$ and $d_1 \leq d \leq 2d_2$ if $2d_2 > d_1$.

7. Let C_i be an $[n, k_i, d_i]$-code over $\mathbb{F}_q$ for $i = 1, 2$. Define

$$C = \{(\boldsymbol{a} + \boldsymbol{x}, \boldsymbol{b} + \boldsymbol{x}, \boldsymbol{a} + \boldsymbol{b} + \boldsymbol{x}) \mid \boldsymbol{a}, \boldsymbol{b} \in C_1, \boldsymbol{x} \in C_2\}.$$

   (a) Show that C is a $[3n, 2k_1 + k_2]$-linear code over $\mathbb{F}_q$.

   (b) If G_i is a generator matrix of C_i, for $i = 1, 2$, find a generator matrix of C in terms of G_1 and G_2.

   (c) If H_i is a parity-check matrix of C_i, for $i = 1, 2$, find a parity-check matrix of C in terms of H_1 and H_2.

8. (a) Find the smallest n such that there exists a binary $[n, 50, 3]$-linear code.

   (b) Find the smallest n such that there exists a binary $[n, 60, 4]$-linear code.

9. Find the smallest n such that there exists an $[n, 40, 3]$-code over $\mathbb{F}_9$.

10. (a) Write down the codewords in $\mathcal{R}(1, m)$ for $m = 3, 4, 5$.

    (b) Verify that $\mathcal{R}(1, 3)$ is self-dual.

11. Show that $\mathcal{R}(r, m)$ has parameters $[2^m, \binom{m}{0} + \binom{m}{1} + \ldots + \binom{m}{r}, 2^{m-r}$.

12. For $0 \leq r < m$ show that $\mathcal{R}(r, m)^{\perp} = \mathcal{R}(m - 1 - r, m)$.

13. Write down the binary solutions of the equation

$$x_1 + x_2 + \ldots + x_m = 1$$

    as column vectors of $\mathbb{F}_2^m$. Let $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ be all the solutions of the above equation. Let C_m be the binary linear code with

$$G = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$$

    as a generator matrix.

(a) Determine all the codewords of C_m for $m = 2, 3, 4$.

(b) Find the parameters of C_m for all m.

14. For a linear code V over $\mathbb{F}_q$, the parameters of V are denoted by

$$\text{length}(V), \dim(V), d(V) = \text{ minimum distance }.$$

Suppose we have

(1) a code C with $\text{length}(C) = n$ and $\dim(C) = k$, and

(2) a collection of k codes $W_1, \ldots, W_k$, all of them having the same length n.

The elemnts of C are written as row vectors, and the elements of W_j are written as column-vectors. We fix a basis $\{\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{C}^{(k)}\}$ of C and denote by G the $k \times m$ matrix whose rows are $\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(k)}$. Thus G is a generator matrix of C. For $1 \leq j \leq k$, we set

$$C_j = \langle \{\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(j)}\} \rangle \subseteq \mathbb{F}_q^m.$$

Then C_j is a q-ary code of length m and dimension j. Moreover

$$C_1 \subset C_2 \subset \ldots \subset C_k = C.$$

Let M be the set conisting of all the $n \times k$ matrices whose jth column belongs to W_j, for all $1 \leq j \leq k$.

(a) Show that M is an $\mathbb{F}_q$-linear space of dimension $\sum_{i=1}^{k} \dim(W_j)$.

(b) If we identify an $n \times m$ matrix A witha vector $\boldsymbol{a}$ of $\mathbb{F}_q^{mn}$ by putting the ith row of A in the ith block of M positions of $\boldsymbol{a}$, then the q-ary linear code

$$W = \{AG \mid A \in M\}$$

has parameters

$$\begin{aligned} \text{length}(W) &= mn, \\ \dim(W) &= \sum_{j=1}^{k} \dim(W_j), \\ d(W) &\geq \min\{d(W_j) \cdot d(C_j) \mid 1 \leq j \leq k\}. \end{aligned}$$

(c) By using the binary codes with parameters $[2, 1, 2]$, $[20, 19, 2]$ and $[20, 14, 4]$, show that we can produce a binary $[40, 33, 4]$-code.

15. Да се докаже, че ако съществува $[N, K, D]$-код над $\mathbb{F}_{q^{n-1}}$, то съществува и $[nN, (n-1)K, 2D]$-код над $\mathbb{F}_q$.

16. Нека α е корен на $1 + x^2 + x^3 \in \mathbb{F}_2[x]$. Да разгледаме изображението

$$\phi : \mathbb{F}_8 \to \mathbb{F}_2^3, \quad a_1 \cdot 1 + a_2 \cdot \alpha + a_3 \cdot \alpha^2 \mapsto (a_1, a_2, a_3).$$

Нека $A = \langle (\alpha+1, \alpha^2+1, 1) \rangle / \mathbb{F}_8$. Намерете всички думи на кода $\phi^*(A) = \{(\phi(c_1), \phi(c_2), \phi(c_3)) \mid (c_1, c_2, c_3) \in A\}$.

17. Да разгледаме линеиония код

$$A := \langle\{(1,1),(\alpha,1+\alpha)\}\rangle$$

над $\mathbb{F}_4$, където α е корен на $1+x+x^2 \in \mathbb{F}_2[x]$. Нека B е двоичният код $\{0000, 1100, 1010, 0110\}$ и да разгледаме $\mathbb{F}_2$-линейно изображение от $\mathbb{F}_4$ в B, дефинирано чрез

$$\phi: \quad 0 \mapsto 0000, \quad 1 \mapsto 1100, \quad \alpha \mapsto 1010, \quad 1+\alpha \mapsto 0110.$$

Намерете всички кодови думи на кода

$$C := \phi^*(A) = \{(\phi(c_1), \phi(c_2)) \mid (c_1, c_2) \in A\}.$$

Глава 5

Циклични кодове

5.1 Основни дефиниции

Един важен клас от кодове е класът на т.нар. циклични кодове. Приема се, че те са изследвани за пръв път от Прейндж през 1957 г. [11]. Много от най-важните линейни кодове като кодовете на Хеминг и Голей, квадратично остатъчните кодове, кодовете на Рид-Соломон и т.н. са циклични. Започваме с дефиниция на цикличните кодове.

Дефиниция 5.1. Един код C наричаме *цикличен*, ако той е линеен и цикличното изместване на коя да е кодова дума е отново кодова дума. С други думи, ако $(a_0, a_1, a_2, \ldots, a_{n-1}) \in C$, то и $(a_{n-1}, a_0, a_1, \ldots, a_{n-2}) \in C$.

Казваме, че думата $(a_{n-r}, \ldots, a_{n-1}, a_0, a_1, \ldots, a_{n-r-1})$ е получена от думата $(a_0, a_1, \ldots, a_{n-1})$ чрез циклично изместване на r позиции. Лесно се проверява, че ортогоналният код на цикличен код е отново цикличен код.

Пример 5.2. Следните кодове са пример за циклични кодове:

(i) тривиалните кододве $\{\mathbf{0}\}$ и $\mathbb{F}_q^n$;

(ii) кодовете с повторение $\{\lambda, \ldots, \lambda) \mid \lambda \in \mathbb{F}_q\}$;

(iii) двоичният $[3, 2, 2]$-код $\{(000, 110, 101, 011\}$;

(iv) двоичният линеен код $\{(0000, 1001, 0110, 1111\}$ не е цикличен, но е еквивалентен на цикличен; резменяйки третата и четвъртата координата получаваме цикличния код $\{(0000, 1010, 0101, 1111\}$;

(v) симплекс-кодът

$$S(3, 2) = \{0000000, 1011100, 0101110, 0010111, 1110010, 0111001, 1001011, 1100101\}.$$

За да получим алгебрично описание на цикличните кодове разглеждаме следното изображение:

$$\pi : \left\{ \begin{array}{ccc} \mathbb{F}_q^n & \to & \mathbb{F}_q[x]/(x^n-1), \\ (a_0, a_1, \ldots, a_{n-1}) & \to & a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}. \end{array} \right. \tag{5.1}$$

Ясно е, че π е $\mathbb{F}_q$-линейно изображение на векторни пространства. По-нататък ще идентифицираме $\mathbb{F}_q^n$ със $\mathbb{F}_q^n[x]/(x^n-1)$ и вектора $(u_0, u_1, \ldots, u_{n-1})$ с полинома $u(x) = u_0 + u_1 x + \ldots + u_{n-1}x^{n-1}$.

Дефиниция 5.3. Нека R е пръстен. Едно непразно подмножество I на R наричаме *идеал*, ако

(i) разликата $a - b$ принадлежит на I за всички $a, b \in I$;

(ii) $ra \in I$ за всяко $r \in R$ и всяко $a \in I$.

Пример 5.4. (i) Всички четни числа в $\mathbb{Z}$ образуват идеал.

(ii) За фиширано цяло положително m целите числа, делящи се на m образуват идеал в $\mathbb{Z}$.

(iii) Нека $f(x)$ е ненулев полином. Всички полиноми, делящи се на $f(x)$ в пръстена от полиноми $\mathbb{F}_q[x]$ образуват идеал.

(iv) Нека $g(x)$ е делител на $x^n - 1$. Всички полиноми в пръстена $\mathbb{F}_q[x]/(x^n-1)$, които се делят на $g(x)$ образуват идеал.

Пример 5.5. Да разгледаме цикличния код $C = \{000, 110, 101, 011\}$. Имаме $\pi(C) = \{0, 1 + x, 1 + x^2, x + x^2\}$. Оказва се, че в пръстена $\mathbb{F}_2[x]/(x^3-1)$ подмножеството $I = \pi(C)$ е идеал.

Дефиниция 5.6. Един идеал I в пръстена R наричаме *главен идеал*, ако съществува елемент $g \in I$ такъв, че $I = \langle g \rangle$, където

$$\langle g \rangle := \{gr \mid r \in R\}.$$

Елементът g се нарича *пораждащ* за I и казваме, че I е породен от g. Един пръстен ще наричаме *област на главни идеали*, ако всеки идеал в R е главен.

Идеалът I отпоследния пример е главен. Лесно се проверява, че $I = \langle 1 + x \rangle$.

Теорема 5.7. Пръстените $\mathbb{Z}$, $\mathbb{F}_q[x]$ и $\mathbb{F}_q[x]/(x^n-1)$ са области на главни идеали.

Доказателство. □

5.2 Пораждащи полиноми

Причината за въвеждане на идеали в предния раздел следната теорема, която свързва идеалите с цикличните кодове.

Теорема 5.8. Нека π е линейното изображение, дефинирано в (5.1). Тогава непразното подмножество C на $\mathbb{F}_q^n$ е цикличен код тогава и само тогава, когато $\pi(C)$ е идеал в $\mathbb{F}_q[x]/(x^n-1)$.

Доказателство. Да допуснем, че $\pi(C)$ е идеал в $\mathbb{F}_q[x]/(x^n-1)$. Тогава за всички $\alpha, \beta \in \mathbb{F}_q \subset \mathbb{F}_q[x]/(x^n-1)$ и всички $\boldsymbol{a}, \boldsymbol{b} \in C$ имаме $\alpha\pi(\boldsymbol{a}), \beta\pi(\boldsymbol{b}) \in \pi(C)$ (съгласно дефиницията на идеал). Следователно $\alpha\pi(\boldsymbol{a}) + \beta\pi(\boldsymbol{b})$ е елемент на $\pi(C)$, т.е. $\pi(\alpha\boldsymbol{a} + \beta\boldsymbol{b}) \in \pi(C)$, откъдето следва, че $\alpha\boldsymbol{a} + \beta\boldsymbol{b}$ е дума от C. Това доказва, че C е линеен код.

Нека сега $\boldsymbol{c} = (c_0, c_1, \ldots, c_{n-1})$ е кодова дума от C. Полиномът

$$\pi(c) = c_0 + c_1x + \ldots + c_{n-2}x^{n-2} + c_{n-1}x^{n-1}$$

е елемент на $\pi(C)$. Тъй като $\pi(C)$ е идеал, то елементът

$$\begin{aligned} x\pi(\boldsymbol{c}) &= c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} + c_{n-1}x^n \\ &= c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} + c_{n-1}(x^n - 1) \\ &= c_{n-1} + c_0x + c_1x^2 + \ldots + c_{n-2}x^{n-1} \end{aligned}$$

е в $\pi(C)$, т.е. $(c_{n-1}, c_0, c_1, \ldots, c_{n-2})$ е кодова дума от C. Това означава, че C е цикличен код.

Обратно, нека предположим, че C е цикличен код. Тогава е ясно, че първото условие от Дефиниция 5.3 се удовлетворява за $\pi(C)$. За всеки полином

$$f(x) = f_0 + f_1x + \ldots + f_{n-2}x^{n-2} + f_{n-1}x^{n-1} = \pi(f_0, f_1, \ldots, f_{n-1})$$

от $\pi(C)$ имаме $(f_0, f_1, \ldots, f_{n-1}) \in C$, а полиномът

$$xf(x) = f_{n-1} + f_0x + f_1x^2 + \ldots + f_{n-2}x^{n-1}$$

е също елемент на $\pi(C)$, тъй като C е цикличен. Отукк следва, че и $x^2f(x) = x(xf(x))$ е елемент на $\pi(C)$. По индукция получаваме, че $x^if(x)$ принадлежи на $\pi(C)$ за всяко $i \geq 0$. Тъй като C е линеен код, а π е линейна трансформация, то $\pi(C)$ е линейно пространство над $\mathbb{F}_q$. Следователно за всеки полином $g(x) = g_0 + g_1x + \ldots + g_{n-1}x^{n-1} \in \mathbb{F}_q[x]/(x^n - 1$, полиномът

$$g(x)f(x) = \sum_{i=0}^{n-1} g_i(x^if(x))$$

е елемент на $\pi(C)$. Следователно, $\pi(C)$ е идеал на $\mathbb{F}_q[x]/(x^n-1)$. □

Пример 5.9. (i) Кодът $C = \{(0,0,0), (1,1,1), (2,2,2)\}$ е троичен цикличен код. Съответният идеал във $\mathbb{F}_3[x]/(x^3-1)$ е $\pi(C) = \{0, 1+x+x^2, 2+2x+2x^2$.

(ii) Множеството $I = \{0, 1+x^2, x+x^3, 1+x+x^2+x^3\}$ е идеал в $\mathbb{F}_2[x]/(x^4-1)$. Съответният цикличен код е $\pi^{-1}(I) = \{0000, 1010, 0101, 1111\}$.

(iii) Тривиалните циклични кодове $\{\mathbf{0}\}$ и $\mathbb{F}_q$ съответстват на тривиалните идеал (0) и $\mathbb{F}_q[x]/(x^n-1)$.

Теорема 5.10. Нека I е ненулев идеал в $\mathbb{F}_q[x]/(x^n-1)$ и нека $g(x)$ е ненулев полином със старши коефицент 1, имащ минимална степен в I. Тогава $g(x)$ е поражда I и дели x^n-1.

Доказателство. Първата част следва от Тхеорем 5.7. По-нататък имаме $x^n-1 = s(x)g(x)+r(x)$ като $\deg(r(x)) < \deg(g(x))$. Оттук следва, че $r(x) = (x^n-1)-s(x)g(x)$ е елемент на I. Тъй като $g(x)$ е от минимална степен $r(x)=0$. Следователно $g(x)$ дели x^n-1. □

Пример 5.11. В Пример **??**(i), полиномът $1+x+x^2$ е от минимална степе. Той дели x^3-1. В Пример **??**(ii), полиномът $1+x^2$ е с най-ниска степен и дели x^4-1. За тривиалния код $\mathbb{F}_q^n$ от най-ниска степен е полиномът 1.

Съгласно Теорема 5.7 всеки идеал в $\mathbb{F}_q[x]/(x^n-1)$ е главен. Така един цикличен код се поражда от кой да е от пораждащите на $\pi(C)$. Обикновено имаме повече от един пораждащ за идеал в $\mathbb{F}_q[x]/(x^n-1)$. Следващият резултат показва, че този пораждащ е единствен, ако наложим някои допълнителни условия.

Теорема 5.12. Във всеки ненулев идеал I на $\mathbb{F}_q[x]/(x^n-1)$ съществува единствен полином от най-ниска степен със старши коефициент 1.

Доказателство. Съгласно Теорема 5.10 такъв полином е пораждащ за I. Ако $g_1(x)$ и $g_2(x)$ са различни пораждащи на I с минимална степен, то подходящо скаларно кратно на $g_1(x)-g_2(x)$ е ненулев полином от I със старши коефициент 1, имащ по-ниска степен от минималната, противоречие. □

Тази теорема оправдава следната дефиниция.

Дефиниция 5.13. Нека I е ненулев идеал в $\mathbb{F}_q[x]/(x^n-1)$. Единственият полином в I от наи-ниска степен и със старши коефицент 1 наричаме *пораждащ полином* за I. За цикличен код C пораждащият полином на идеала $\pi(C)$ наричаме *пораждащ полином на кода* C.

Пример 5.14. (i) Пораждащият полином на цикличния код $\{000.110, 101, 011\}$ е $1+x$.

(ii) Пораждащия полином на симплекс кода от Пример 5.2 е $1+x^2+x^3+x^4$.

Теорема 5.15. Всеки делител на x^n-1, имащ старши коефициент 1 е пораждащ полином на нчкакъв цикличен код над $\mathbb{F}_q$.

Доказателство. Нека $g(x)$ е делител на x^n-1 със старши коефициент 1 и нека $I = \langle g(x)\rangle \lhd \mathbb{F}_q/(x^n-1)$. Нека C е съответният цикличен код и да означим с $h(x)$ пораждащия полином на C. Тогава съществува полином $b(x)$, за който

$$h(x) \equiv g(x)b(x) \pmod{x^n-1}.$$

Оттук следва, че $g(x)$ е делител на $h(x)$ и тъй като $h(x)$ е от минмална степен в I и е със старши коефициент 1, то $g(x) \equiv h(x)$. □

От Теореми 5.12 и 5.15 получаваме следния резултат.

Следствие 5.16. Съществува взаимно-еднозначно съответствие между цикличните кододве във $\mathbb{F}_q^n$ и делиетлите на $x^n-1 \in \mathbb{F}_q[x]$ със старши коефициент 1.

Очевидно полиномите 1 и $x^n - 1$ са пораждащи полиноми на тривиалните кодове $\mathbb{F}_q^n$ и $\mathbf{0}$.

Пример 5.17. Ще опишем всички двоични циклични кодове с дължина 6. да разложим на множители $x^6 - 1$:

$$x^6 - 1 = (x^2 + 1)(x^4 + x^2 + 1) = (x + 1)^2(x^2 + x + 1)^2.$$

Оттук получаваме девет делителя на $x^6 - 1$ със старши коефициент 1:

$$\begin{array}{ccc} 1, & 1 + x, & (1 + x)^2 \\ 1 + x + x^2, & (1 + x)(1 + x + x)^2, & (1 + x)^2(1 + x + x^2) \\ (1 + x + x^2)^2, & (1 + x + x^2)^2(1 + x), & (1 + x + x^2)^2(1 + x)^2 \end{array}$$

Съществуват девет девет двоични циклични кода с дължина 6. Лесно могат да се напишат кодовите думи за всеки от тези кодове. Така например, кодът, породен от $(1 + x + x^2)^2$ съдържа думите

$$\{000000, 101010, 010101, 111111\}$$

От горния пример е ясно, че броят на цикличните кодове с дължина n може да бъде определен лесно, ако знаем разлагането на $x^n - 1$ над $\mathbb{F}_q$ на прости множители.

Теорема 5.18. Нека $x^n - 1 \in \mathbb{F}_q[x]$ има следното разлагане на прости множители над $\mathbb{F}_q$

$$x^n - 1 = \prod_{[} i = 1]^r p_i^{e_i}(x),$$

където $p_1(x), \ldots, p_r(x)$ са различни неразложими полиноми със старши коефициент 1, а $e_i \geq 1$ за $i = 1, \ldots, r$. Тогава съществуват $\prod_{i=1}^{r}(e_i + 1)$ с дължина n над $\mathbb{F}_q$.

В таблиците по-долу е представено разлагането на полинома $x^n - 1$ над $\mathbb{F}_q$ за $q = 2, 3$ и дължини $n \leq 10$ и е пресметнат броя на съответните циклични кододове.

n	разлагане на $x^n - 1$	# циклични кодове
1	$1 + x$	2
2	$(1 + x)^2$	3
3	$(1 + x)(1 + x + x^2)$	4
4	$(1 + x)^4$	5
5	$(1 + x)(1 + x + x^2 + x^3 + x^4)$	4
6	$(1 + x)^2(1 + x + x^2)^2$	9
7	$(1 + x)(1 + x + x^3)(1 + x^2 + x^3)$	8
8	$(1 + x)^8$	9
9	$(1 + x)(1 + x + x^2)(1 + x^3 + x^6)$	8
10	$(1 + x)^2(1 + x + x^2 + x^3 + x^4)^2$	9

n	разлагане на x^n-1	# циклични кодове
1	$1+x$	2
2	$(2+x)(1+x)$	4
3	$(2+x)^3$	4
4	$(2+x)(1+x)(1+x^2)$	8
5	$(2+x)(1+x+x^2+x^3+x^4)$	4
6	$(2+x)^3(1+x)^3$	16
7	$(2+x)(1+x+x^2+x^3+x^4+x^5+x^6)$	4
8	$(2+x)(1+x)(1+x^2)(2+x+x^2)(2+2x+x^2)$	32
9	$(2+x)^9$	10
10	$(2+x)(1+x)(1+x+x^2+x^3+x^4)(1+2x+x^2+2x^3+x^4)$	16

Един цикличен код се определя напълно от пораждащия полином. Следователно всички параметри на цикличен код също се определят от пораждащия полином. Следващата теорема показва как размерността на един цикличен код зависи от пораждащия полином.

Теорема 5.19. Нека $g(x)$ е пораждащият полином на идеал на $\mathbb{F}_q/(x^n-1)$ и нека $\deg g(x) = n-k$. Тогава цикличният код, породен от $g(x)$ е с размерност k.

Доказателство. Нека $c_1(x) \neq c_2(x)$ са полиноми с $\deg c_i(x) \leq k-1$, $i=1,2$. Тогва $g(x)c_1(x) \not\equiv g(x)c_2(x) \pmod{x^n-1}$. Следователно множеството

$$A := \{g(x)c(x) \mid c(x) \in \mathbb{F}_q[x]/(x^n-1), \deg c(x) \leq k-1\}$$

има q^k елемента и е подмножество на идеала $\langle g(x)\rangle$. От друга страна за всяка кодова дума $g(x)a(x)$ със $a(x) \in \mathbb{F}_q[x]/(x^n-1)$ имаме

$$a(x)g(x) = u(x)(x^n-1) + v(x), \tag{5.2}$$

където $\deg v(x) < n$. От (5.2) имаме $v(x) = a(x)g(x) - u(x)(x^n-1)$. Следователно $g(x)$ дели $v(x)$. Да представим $v(x)$ във вида $v(x) = b(x)g(x)$ за някакъв полином $b(x)$. Тогава $\deg b(x) < k$ и $v(x)$ е в A. Това показва, че $A = \langle g(x)\rangle$. Следователно фразмерността на кода е $\log_q |A| = k$. □

Пример 5.20. (i) От разлагането $x^7-1 = (1+x)(1+x^2+x^3)(1+x+x^3) \in \mathbb{F}_2[x]$ получаваме, че съществуват два двоични $[7,3]$-кода:

$$\begin{aligned}\langle(1+x)(1+x^2+x^3)\rangle &= \{0000000, 1110100, 0111010, 0011101, 0010111\\ &\quad 1001110, 0100111, 1010011, 1101001\}\end{aligned}$$

и

$$\begin{aligned}\langle(1+x)(1+x+x^3)\rangle &= \{0000000, 1011100, 0101110, 0010111\\ &\quad 1001011, 1100101, 1110010, 0111001\}.\end{aligned}$$

(ii) От разлагането

$$x^7-1 = (2+x)(1+x+x^2+x^3+x^4+x^5+x^6+x^7)$$

получаваме, че не съществуват двоични циклични $[7,2]$-кодове.

5.3 Пораждаща и проверочна матрица

В предния раздел показахме, че един цикличен код напълно се определя от пораждащия си полином. Следователно той трябва да определея и порахздаща матрица за кода. Следващата теорема установява връзка между тези два обекта.

Теорема 5.21. Нека $g(x) = g_0 + g_1 x + \ldots + g_{n-k}x^{n-k}$, $g_{n-k} \neq 0$, е пораждащичт полином на цикличен код C над $\mathbb{F}_q$. Тогава матрицата

$$G = \begin{pmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & & \ldots & g_{n-k} & 0 & 0 & 0 & \ldots & 0 \\ 0 & g_0 & g_1 & \ldots & & g_{n-k} & 0 & 0 & \ldots & 0 \\ \vdots & & & \ddots & & & & & \ddots & \vdots \\ 0 & 0 & & \ldots & g_0 & g_1 & & & \ldots & g_{n-k} \end{pmatrix}$$

е пораждаща матрица на кода C.

Доказателство. Достатъчно е да се покаже, че $g(x), xg(x), \ldots, x^{k-1}g(x)$ образуват базис на C. Ясно е, че те са линейно независими над $\mathbb{F}_q$. От Теорема 5.19 имаме, еч $\dim C = k$, откъдето следва желаният резултат. □

Пример 5.22. Да разгледаме двоичния $[7, 4]$-код с пораждащ полином $g(x) = 1 + x^2 + x^3$. Този код има пораждаща матрица

$$G = \begin{pmatrix} g(x) \\ xg(x) \\ x^2g(x) \\ x^3g(x) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

Тази пораждаща матрица не е в стандартна форма. Ако към втория ред добавим четвъртия, а към първия добвим сумата на последните два, получаваме

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

Тази матрица е в стандартна форма и от нея лесно можем да получим проверочна матрица за нашия код:

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

□

Известно е, че проверочна матрица на даден линеен код може да бъде получена от коя да е негова пораждаща матрица. Тъй като ортогоналният на цикличен код е също цикличен, то е възможно да намерим проверочна матрица направо от пораждащия полином на ортогоналния код. Задачата се свежда до намиране на пораждащ полином за ортогоналния код $C^{\perp}$.

Дефиниция 5.23. Нека $h(x) = \sum_{i=0}^{k} a_i x^i$ е полином от степен k, $a_k \neq 0$, над $\mathbb{F}_q$. Дефинираме *реципрочния полином* $h^{(R)}(x)$ на $h(x)$ чрез

$$h^{(R)}(x) = x^k h(1/x) = \sum_{i=0}^{k} a_{k-i} x^i.$$

От дефиницията веднага следва, че ако $h(x)$, дели $x^n - 1$, то и $h^{(R)}(x)$ дели $x^n - 1$.

Пример 5.24. (i) Ако $h(x) = 1 + 2x + 3x^5 + x^7 \in \mathbb{F}_5[x]$, то реципрочният му е

$$\begin{aligned} h^{(R)}(x) &= x^7 h(1/x) \\ &= x^7(1 + 2(1/x) + 3(1/x)^5 + (1/x)^7) \\ &= 1 + 3x^2 + 2x^6 + x^7. \end{aligned}$$

(ii) Да разгледаме полинома $h(x) = 1+x+x^3 \in \mathbb{F}_2[x]$, който е делител на x^7-1. Тогава $h^{(R)}(x) = 1 + x^2 + x^3$ е също делител на $x^7 - 1$. □

Пример 5.25. Нека $g(x) = g_0 + g_1 x + g_2 x^2 + g_3 x^3$ е пораждащият полином на цикличен код над $\mathbb{F}_q$ с дължина 4 и нека $h(x) = (x^n - 1)/g(x)$. Да положим $h(x) = h_0 + h_1 x + h_2 x^2 + h_3 x^3$. Тогава $h^{(R)}(x) = (h_3 + h_2 x + h_1 x^2 + h_0 x^3)/x^{3-k}$, където $k = \deg h(x)$. Да разгледаме произведението

$$\begin{aligned} 0 &\equiv g(x)h(x) \\ &\equiv (g_0 + g_1 x + g_2 x^2 + g_3 x^3)(h_0 + h_1 x + h_2 x^2 + h_3 x^3) \\ &\equiv g_0 h_0 + (g_0 h_1 + g_1 h_0)x + (g_0 h_2 + g_1 h_1 + g_2 h_0)x^2 + \\ &\quad (g_0 h_3 + g_1 h_2 + g_2 h_1 + g_3 h_0)x^3 + (g_1 h_3 + g_2 h_2 + g_3 h_1)x^4 + \\ &\quad (g_2 h_3 + g_3 h_2)x^5 + g_3 h_3 x^6 \\ &\equiv (g_0 h_0 + g_1 h_3 + g_2 h_2 + g_3 h_1) + (g_0 h_1 + g_1 h_0 + g_2 h_3 + g_3 h_2)x + \\ &\quad (g_0 h_2 + g_1 h_1 + g_2 h_0 + g_3 h_3)x^2 + (g_0 h_3 + g_1 h_2 + g_2 h_1 + g_3 h_0)x^3 \pmod{x^4 - 1}. \end{aligned}$$

В него коефициентът пред всяка от степените на x трябва да бъде 0.

Да положим $\boldsymbol{b} = (h_3, h_2, h_1, h_0) \in \mathbb{F}_q^4$ и $\boldsymbol{g} = (g_0, g-1, g_2, g_3) \in \mathbb{F}_q^4$. Да означим с $\boldsymbol{g}_i$ вектора получен от $\boldsymbol{g}$ чрез циклично изместване на i позиции. Сравнявайки коефициентите пред x^3 в горното сравнение получаваме

$$\boldsymbol{g}_0 \cdot \boldsymbol{b} = \boldsymbol{g} \cdot \boldsymbol{b} = g_0 h_3 + g_1 h_2 + g_2 h_1 + g_3 h_0 = 0.$$

Въобще сравнявайки степените пред x^i, $i = 0, 1, 2, 3$, получаваме $\boldsymbol{g}_i \cdot \boldsymbol{b} = 0$. Тъй като C се поражда от $\{\boldsymbol{g}_0, \boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3$, то $\boldsymbol{b}$ е кодова дума от $C^\perp$. Изместваики циклично вектора $\boldsymbol{b} = (h_3, h_2, h_1, h_0)$ на $k+1$ позиции ще получим вектора, съответстващ на $h^{(R)}(x)$. Тъй като $C^\perp$ е цикличен, то и $h^{(R)}(x)$ е кодова дума.

По дефиниция имаме $\deg h^{(R)}(x) = \deg h(x) = k$. Следователно множеството от $\{h^{(r)}(x), xh^{(r)}(x), \ldots, x$ е базис на $C^\perp$. Така получихме, че $C^\perp$ се поражда от $h_0^{-1} h^{(R)}(x)$. (Да отбележим, че $h_0 = h(0) \neq 0$, тъй като $h_0 g_0 = h(0)g(0) = -1$.) □

Ясно е, че Пример 5.25 може да бъде обобщен за циклични кодове с произволна дължина n. Това е съдържанието на следващата теорема.

Теорема 5.26. Нека $g(x)$ е пораждащия полином на q-ичен цикличен $[n,k]$-код C и нека $h(x) = (x^n - 1)/g(x)$. Тогава $h_0^{-1}h^{(R)}(x)$, където h_0 е свободният член на $h(x)$ е пораждащият полином на ортогоналния код $C^\perp$.

Доказателство. Нека $g(x) = \sum_{i=0}^{n-1} g_i x^i$ и $h(x) = \sum_{i=0}^{n-1} h_i x^i$. Тогава

$$\begin{aligned} 0 &\equiv g(x)h(x) \\ &\equiv (g_0h_0 + g_1h_{n-1} + \ldots + g_{n-1}h_1) + (g_0h_1 + g_1h_0 + \ldots + g_{n-1}h_2)x + \\ &\quad (g_0h_2 + g_1h_1 + \ldots + g_{n-1}h_3)x^2 + \ldots + \\ &\quad (g_0h_{n-1} + g_1h_{n-2} + \ldots + g_{n-1}h_0)x^{n-1} \pmod{x^n - 1}. \end{aligned}$$

Коефициентите пред степените на x в горния израз трябва да бъдат 0. Ако означим с $\boldsymbol{g}_i$ вектора, получен от $(g_0, g_1, \ldots, g_{n-1})$ чрез циклично изместване на i позиции, $i = 0, 1, \ldots, n-1$, то получаваме

$$\boldsymbol{g}_i \cdot (h_{n-1}, h_{n-2}, \ldots, h_1, h_0), \;\; i = 0, 1, \ldots, n-1.$$

Тъй като $\{\boldsymbol{g}_0, \boldsymbol{g}_1, \ldots, \boldsymbol{g}_{n-1}\}$ пораждат кода C (съгласно Теорема 5.21), то $(h_{n-1}, h_{n-2}, \ldots, h-1, h_0)$ е кодова дума от $C^\perp$.

Като изместим циклично вектора $(h_{n-1}, h_{n-2}, \ldots, h-1, h_0)$ на $k = 1$ позиции получаваме вектора, съответстващ на $h^{(R)}(x)$. Оттук следва, че $h^{(R)}(x)$ също е кодова дума, тъй като и кодът $C^\perp$ е цикличен. По-нататък имаме $\deg h(x) = \deg h^{(R)}(x) = k$, откъдето следва, че множеството $\{h^{(R)}(x), xh^{(R)}(x), \ldots, x^{n-k-1}h^{(R)}(x)\}$ е базис на $C^\perp$. Следователно $C^\perp$ се поражда от $h^{(R)}(x)$ и полиномът $h_0^{-1}h^{(R)}(x)$ състарши коефициент 1 е пораждащ полином на $C^\perp$. □

Дефиниция 5.27. Нека C е q-ичен код с дължина n и нека $h(x) = (x^n-1)/g(x)$. Полинома $h_0^{-1}h^{(R)}(x)$, в който h_0 е свободният член на $h(x)$ наричаме *проверочен полином на* C.

Следствие 5.28. Нека C е q-ичен цикличен $[n,k]$-код с пораждащ полином $g(x)$ и нека $h(x) = (x^n - 1)/g(x)$. Да положим $h(x) = h_0 + h_1x + \ldots + h_kx^k$. Тогава матрицата

$$H = \begin{pmatrix} h^{(R)}(x) \\ xh^{(R)}(x) \\ \vdots \\ x^{n-k-1}h^{(R)}(x) \end{pmatrix} = \begin{pmatrix} h_k & h_{k-1} & & \ldots & h_0 & 0 & 0 & 0 & \ldots & 0 \\ 0 & h_k & h_{k-1} & \ldots & & h_0 & 0 & 0 & \ldots & 0 \\ \vdots & & & \ddots & & & & & \ddots & \vdots \\ 0 & 0 & & \ldots & h_k & h_{k-1} & & & \ldots & h_0 \end{pmatrix}$$

е проверочна матрица на C.

Пример 5.29. Нека C е двоичен цикличен $[7,4]$-код, породен от полинома $g(x) = 1+x^2+x^3$. Полагаме $h(x) = (x^7 - 1)/g(x) = 1 + x^2 + x^3 + x^4$. Тогава $h^{(R)}(x) = 1 + x + x^2 + x^4$ е проверочният полином на C. Следователно,

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

е проверочна матрица на C.

5.4 Декодиране на циклични кодове

Декодирането на циклични кодове се свежда до извършването на същите три стъпки, които правим при декодирането на общи линейни кодове: пресмятане на синдрома, намиране на вектор-грешка, който му съответства и поправяне на грешките. Поради богатата структура на цикличните кодове и разнообразните алгебрични и геометрични свойства, коио те притежават тези стъпки са обикновено много леки и ние можем да достигнем простота и ефективност на декодирането.

Използвайки Следствие 5.28, ние лесно можем да получим чрез елементарни операции по редове проверочна матрица във вида

$$H = (I_{n-k}|A). \tag{5.3}$$

За всеки циклчен код C проверочна матрица от вида (5.3) съществува и е единствена. Всички синдроми в този раздел се пресмятат по отношение на проверочна матрица от вида (5.3).

Теорема 5.30. Нека $H = (I_{n-k}|A)$ е проверочна матрица на q-ичен цикличен код C и нека $g(x)$ е пораждащият полином на C. Тогава синдромът на вектора $\boldsymbol{w} \in \mathbb{F}_q^n$ е равен на $(w(x) \pmod{g(x)})$, т.е. остатъка получен при делението на $w(x)$ на $g(x)$. (Тук идентифицираме вектора $\boldsymbol{w} \in \mathbb{F}_q^n$ с полинома $w(x) \in \mathbb{F}_q[x]/(x^n - 1)$.)

Доказателство. Матрицата A е с $n-k$ реда и k стълба. С всеки стълб на A свързваме полином от степен, ненадминаваща $n-k-1$, и записваме A във вида

$$A = (a_0(x), a_1(x), \ldots, a_{k-1}(x)).$$

Известно е, че $G = (-A^T|I)$ е пораждаща матрица на C. Следователно $x^{n-k+i} - a_i(x)$ е кодова дума от C. Да положим $x^{n-k+i} - a_i(x) = q_i(x)g(x)$ за някакъв полином $q_i(x) \in \mathbb{F}_q[x]/(x^n - 1)$, откъдето получаваме

$$a_i(x) = x^{n-k+i} - q_i(x)g(x).$$

Да положим $w(x) = w_0 + w_1x + \ldots + w_{n-1}x^{n-1}$. Полиномът $s(x)$, съответстващ на синдрома $\boldsymbol{s} = H\boldsymbol{w}$ е

$$\begin{aligned} s(x) &= w_0 + w_1x + \ldots + w_{n-k-1}x^{n-k-1} + w_{n-k}a_0(x) + \ldots + w_{n-1}a_{k-1}(x) \\ &= \sum_{i=0}^{n-k-1} w_ix^i + \sum_{j=0}^{k-1} w_{n-k+j}(x^{n-k+j} - q_j(x)g(x)) \\ &= \sum_{i=0}^{n-1} w_ix^i + \left(\sum_{j=0}^{k-1} w_{n-k+j}q_j(x)\right) g(x) \\ &\equiv w(x) \pmod{g(x)}. \end{aligned}$$

Тъй като степента на полинома $s(x)$ не надминава $n-k-1$, получаваме желания резултат. □

Пример 5.31. Да разгледаме двоичен $[7, 4, 3]$0-код на Хеминг с пораждащ полином $g(x) = 1 + x^2 + x^3$. Извършвайки елементарни преобразувания по редове на матрицата от Пример **??** получаваме проверочна матрица $H = (I_3|A)$, където

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

Синдромът на думата $\boldsymbol{w} = 0110110$ е $\boldsymbol{s} = \boldsymbol{w}H^T = 010$. От друга страна,

$$w(x) = x + x^2 + x^4 + x^5 = x + x^2 g(x).$$

Така остатъкът $(w(x) \pmod{g(x)})$ е x, което съответства на думата 010. □

Теорема 5.30 показва, че синдромът на получената дума $w(x)$ може да се определи от остатъка $s(x) = (w(x) \pmod{g(x)})$. Следователно, $w(x) - s(x)$ е кодова дума.

Следствие 5.32. Нека $g(x)$ е пораждащият полином на цикличен од C. Ако $w(x)$ е получената дума, то остатъкът $s(x)$, получен от деленето на $w(x)$ на $g(x)$ е стегло $\leq \lfloor (d(C) - 1)/2 \rfloor$, то тогава $s(x)$ е грешката за $w(x)$, т.е. $w(x)$се декодира в $w(x) - s(x)$ чрез алгоритъма за декодиране по принципа на максималното правдоподобие. □

Доказателство. От Теорема 5.30 следва, че $w(x)$ и $s(x)$ са в един и същи съседен клас. Нека $s(x)$ е лидерът на съседен клас, за който $s(x)\lfloor (d-1)/2 \rfloor$. Оттук следва резултатът. □

Пример 5.33. В Пример 5.31 видяхме, че остатъкът на $w(x) = x+x^2+x^4+x^5$ при делене на $g(x) = 1+x^2+x^3$ е x. Следователно $w(x)$ се декодира във $w(x)-x = x^2+x^4+x^5 = 0010110$. Ако сме получили думата $w_1(x) = 1 + x^2 + x^3 + x^4$, то остатъкък $(w_1(x) \pmod{g(x)})$ е $1 + x + x^2$. В този случай можем да изолзваме синдромно декодиране за да получим думата $w_1(x) - x^4 = 1 + x^2 + x^3 = 1011000$, тъй като думаата 0000100 е лидер на съседния клас, на който принадлежи $w_1(x)$.

От горния пример се вижда, че за някои получени думи можем дадекодираме като веднага извадим остатъка от думата, докато за други трябва да използваме синдромно декодиране. Алгебричните и геометричните свойства на цикличните кодове позволяват да упростим синдромното декодиране за някои получени думи. По-нататък ще опишем такова декодиране, което се нарича *улавяне на грешки* (error trapping).

Лема 5.34. Нека C е q-ичен, цикличен $[n, k]$-код с пораждащ полином $g(x)$. Нека $s(x) = \sum_{i=0}^{n-k-1} s_i x^i$ е синдромът на $w(x)$. Тогава синдромът на цикличния шифт $xw(x)$ е равен на $xs(x) - s_{n-k-1}g(x)$.

Доказателство. От Теорема 5.30 следва, че е достатъчно да покажем, че $xs(x) - s_{n-k-1}g(x)$ е остатъкът на $xw(x)$ при делене на $g(x0$. Нека $w(x) = q(x)g(x)+s(x)$. Тогава

$$xw(x) = xq(x)g(x) + xs(x) = (xq(x) + s_{n-k-1})g(x) + (xs(x) - s_{n-k-1}g(x)).$$

Остава да забележим, че $\deg(xs(x) - s_{n-k-1}g(x)) < n - k = \deg(g(x))$, откъдето следва желания резултат. □

Забележка 5.35. Синдромът на цикличния шифт $x^i w(x)$ на думата $w(x)$ може да се получи от синдрома на $x^{i-1}w(x)$. Така можем да пресметнем индуктивно синдромите на $w(x), xw(x), x^2w(x), \ldots$.

Пример 5.36. От Пример 5.31 знаем, че синдромът на $w(x) = x + x^2 + x^4 + x^5$ е x. Така синдромите на $xw(x)$ и $x^2w(x)$ са съответно $x \cdot x = x^2$ и $x \cdot x^2 - g(x) = 1 + x^2$.

Дефиниция 5.37. Една n-орка от l циклично последователни нули в дадена дума наричаме *цикличен блок от нули с дължина* l.

Пример 5.38. Векторът $\boldsymbol{w} = (1, 3, 0, 0, 0, 0, 0,)$ има цикличен блок от нули с дължина 5, а $\boldsymbol{w} = (0, 0, 1, 2, 0, 0, 0, 1, 0, 0)$ има цикличен блок от нули с дължина 4.

Алгоритъм за декодиране на циклични кодове

Алгоритъм 4. *Вход:* Цикилчен $[n, k, d]$-код C над $\mathbb{F}_q$ с пораждащ полином $g(x)$; получна дума $w(x)$, за която грешката $e(x)$ е с тегло $w_{\mathrm{Ham}}((e(x)) \leq \lfloor (d-1)/2 \rfloor$ и която има цикличен блок от нули с дължина поне k.
Изход: Грешката $e(x)$.

(1) Пресмятат се синдромите на $x^i w(x)$ за $i = 0, 1, 2, \ldots$ и означаваме със $s_i(x)$ синдрома $(x^i w(x) \pmod{g(x)})$.

(2) Намира се такова цяло m, за което синдромът $s_m(x)$ за $x^m w(x)$ е по-малък или равен на $\lfloor (d-1)/2 \rfloor$.

(3) Пресмята се остатъкът $e(x)$ на $x^{n-m} s_m(x)$ при делене на $x^n - 1$. Декодираме $w(x)$ в $w(x) - e(x)$.

Доказателство. Най-напред ще докажем съществуването на числото m от стъпка 2. Съгласно допускането, съществува полином-грешка $e(x)$, който има цикличен блок от нули с дължина поне k. Следователно съществува цяло число $m \geq 0$, за което цикличният шифт на $e(x)$ на m позиции има всичките си ненулеви компоненти в първите $n-k$ позиции. Цикличният шифт на $e(x)$ на m позиции е всъщност остатъкът на $x^m w(x) \pmod{x^n - 1}$ при делене с $g(x)$. Да положим

$$r(x) := ((x^m w(x) \pmod{x^n - 1}) \bmod g(x)) = (x^m w(x) \pmod{g(x)}).$$

Теглото на $r(x)$ същпада с теглото на $e(x)$, което е най-много $\lfloor (d-1)/2 \rfloor$. Това доказва съществуването на m.

Думата $t(x) := (x^{n-m} s_m(x) \pmod{x^n - 1})$ е цикличен шифт на $(\boldsymbol{s}_m, \boldsymbol{0})$ на $n-m$ позиции, където $\boldsymbol{s}_m$ е вектор от $\mathbb{F}_q^{n-k}$, съответстващ на полинома $s_m(x)$. Ясно е, че теглото на $t(x)$ е същото като теглото на $s_m(x)$. Следователно $w_{\mathrm{Ham}}(t(x)) \leq \lfloor (d-1)/2 \rfloor$. Тъй като

и x^m е взаимнопросто с $g(x)$, твърдим, че $w(x) - t(x)$ се дели на $g(x)$, т.е. $w(x) - t(x)$ е кодова дума. Тъй като $t(x)$ и грешката $e(x)$ са в един и същ съседен клас, имаме $e(x) = t(x) = (x^{n-m} s_m(x) \pmod{x^n - 1}$ (от единствеността на лидерите с тегло ненадхвърлящо $\lfloor (d-1)/2 \rfloor$). □

Пример 5.39. (i) Нека е получена думата

$$w_1(x) = 1011100 = 1 + x^2 + x^3 + x^4.$$

Да пресмятаме синдромите $s_i(x)$ на $x^i w_1(x)$ докато $w_{\mathrm{Ham}}(s_i(x)) \leq 1$ (вж. таблицата).

i	s_i
0	$1 + x + x^2$
1	$1 + x$
2	$x + x^2$
3	1

Декодираме $w_1(x) = 1011100$ във

$$w_1(x) - x^4 s_3(x) = w_1(x) - x^4 = 1 + x^2 + x^3 = 1011000.$$

(ii) Да разгледаме двоичния цикличен $[15, 7]$-код, породен от $g(x) = 1 + x^+ x^6 + x^7 + x^8$. Лесно се проверява (да речем от коя да е проверочна матрица), че минималното разстояние на този код е 5. Вектор грешка с тегло не по-голямо от 2 има цикличен блок от нули с дължина поне 7. Следователно ние можем да поправим такъв вектор-грешка с горния алгоритъм. Нека сме получили думата

$$w_2(x) = 110011101100010 = 1 + x + x^4 + x^5 + x^6 + x^8 + x^9 + x^{13}$$

и да пресметнем синдромите $s_i(x)$ на $x^i w_2(x)$ докато получим $w_{\mathrm{Ham}}(s_i(x)) \leq 2$ (вж. таблицата).

i	s_i
0	$1 + x^2 + x^5 + x^7$
1	$1 + x + x^3 + x^4 + x^7$
2	$1 + x + x^2 + x^5 + x^6 + x^7$
3	$1 + x + x^2 + x^3 + x^4$
4	$x + x^2 + x^3 + x^4 + x^5$
5	$x^2 + x^3 + x^4 + x^5 + x^6$
6	$x^3 + x^4 + x^5 + x^6 + x^7$
7	$1 + x^5$

Думата $w_2(x) = 11001101100010$ във

$$w_2(x) - x^8 s_7(x) = w(x) - x^8 - x^{13} = 1 + x + x^4 + x^5 + x^6 + x^9 = 110011100100000.$$

5.5 Кодове, поправящи пакети грешки

Дотук разглеждахме кодове, поправящи случайни грешки. В някои канали за предаване на данни, като например телефонни линии или магнитни запаметяващи устройства, грешките се локализират не по случаен начин, а в къси интервали. Такива грешки наричаме *пакетни грешки* или *пакети от грешки*. В общия случай кодове, поправящи случайни грешки не са подходящи за поправяне на пакети от грешки. Желателно е да се конструират кодове, поправящи точно такива грешки. Въпросните кодове наричаме *кодове, поправящи пакети от грешки*.

Оказва се, че цикличните кодове са много ефективни за коригиране на пакети от грешки. В този раздел ще разгледаме някои свойства на кодовете, поправящи пакети от грешки и ще представим един алгоритъм за декодиране. Всички кодове втози раздел ще бъдат двоични кодове.

Дефиниция 5.40. *Пакет* с дължина $l > 1$ е двоичен вектор, чиито ненулеви компоненти заемат l циклично последователни позиции, при което първата и последната позиции са ненулеви. Един код наричаме *код, поправящ пакет с дължина* l, ако той може да поправя всички грешки, които са пакети с дължина l или по-малко, т.е. всички грешки, които са пакети с дължина l или по-малко.

Пример 5.41.

Теорема 5.42. Един линеен код C е код поправящ всички пакети с дължина l, тогава и само тогава, когато всички грешки, които са пакети с дължина l или по-малко лежат в различни съседни класове на C.

Доказателство.

Следствие 5.43. Нека C е $[n, k]$-код, поправящ пакети с дължина l. Тогава

(i) никой ненулев пакет с дължина $2l$ или по-малко не може да е кодова дума;

(ii) (граница на Райгер) $n - k \geq 2l$.

Доказателство.

За всеки линеен $[n, k]$-код, поправящ пакети с дължина l е изпълненo$n - k \geq 2l$, т.е. $l \leq \lfloor (n - k)/2 \rfloor$. Линеен код, поправящ пакети от грешки и който достига границата на Ригер се нарича *оптимален код, поправтащ пакети от грешки*.

Пример 5.44.

Алгоритъм за декодиране на циклични кодове, поправящи пакети от грешки

Пример 5.45.

5.6 Задачи

1. Определете, кои от следните кодове са циклични.
   (a) $\{(0,0,0),(1,1,1),(2,2,2)\} \subset \mathbb{F}_3^3$;
   (b) $\{(0,0,0),(1,0,0),0,1,0),(0,0,1)\} \in \mathbb{F}_q^3$;
   (c) $\{x_0,x_1,\ldots,x_{n-1} \in \mathbb{F}_q^n \mid \sum_{i=0}^{n-1} x_i = 0\}$;
   (d) $\{x_0,x_1,\ldots,x_{n-1} \in \mathbb{F}_8^n \mid \sum_{i=0}^{n-1} x_i^2 = 0\}$;
   (e) $\{x_0,x_1,\ldots,x_{n-1} \in \mathbb{F}_2^n \mid \sum_{i=0}^{n-1} (x_i^2 + x_i) = 0\}$.
2. Да се докаже, че ортогоналният код на цикличен код е цикличен.
3. Да се докаже, че множеството $I = \{f(x) \in \mathbb{F}_q[x] \mid f(0) = f(1) = 0\}$ е идеал в$\mathbb{F}_q[x]$ и да се намери пораждащ елемент.
4. Нека x и y са две независими променливи. Да се докаже, че пръстенът от полиноми $\mathbb{F}_q[x,y]$ не е област на главни идеали.
5. да се намерят всички пораждащи полиноми със старши коефициент 1 за всеки от следните идеали:
   (a) $I = \langle 1 + x + x^3 \rangle \subset \mathbb{F}_2[x]/(x^7 - 1)$;
   (b) $I = \langle 1 + x^2 \rangle \subset \mathbb{F}_3[x]/(x^4 - 1)$;
6. Да се определи дали следните полиноми са пораждащи за цикличните кодове със зададените дължини:
   (a) $g(x) = 1 + x + x^2 + x^3 + x^4$ за двоичен цикличен код с дължина 7;
   (b) $g(x) = 2 + 2x^2 + x^3$ за троичен цикличен код с дължина 7;
   (c) $g(x) = 2 + 2x + x^3$ за троичен цикличен код с дължина 13.

Глава 6

Специални класове циклични кодове

6.1 БЧХ-кодове

Класът на БЧХ-кодовете е открит независимо от Хоквингем [8] и Боуз и Рей-Чоудхури [1]. Името на класа е съкращение от инициалите на неговите откриватели. Този клас обобщава на кодовете на Хеминг като кодовете в него откриват повече от една грешки. По-нататъшни обобщения на БЧХ-кодовете са направени от Д. Горенстейн и Н. Цирлер [**?**].

6.1.1 Дефиниция на БЧХ-кодове

Да припомним, че *най-малкото общо кратно* на t ненулеви полинома $f_1(x), f_2(x), \ldots, f_t(x) \in \mathbb{F}_q[x]$ е полином от минимална степен със старши коефициент 1, който се дели на всеки от полиномите $f_1(x), f_2(x), \ldots, f_t(x)$. Наи-малкото общо картно на тези полиноми означаваме с $\mathrm{lcm}(f_1(x), \ldots, f_t(x))$. Лесно се показва, че $\mathrm{lcm}(f_1(x), \ldots, f_t(x), f_{t-1}(x)) = \mathrm{lcm}(\mathrm{lcm}(f_1(x), \ldots, f_t(x)), f_t(x))$. Ако за $f_1(x), \ldots, f_t(x)$ имаме разлаганията

$$f_1(x) = a_1 p_1(x)^{e_{1,1}} \ldots p_n(x)^{e_{1,n}}, \ldots, f_t(x) = a_t p_1(x)^{e_{t,1}} \ldots p_n(x)^{e_{t,n}}.$$

където $a_1, \ldots, a_t \in \mathbb{F}_q^*$, $e_{i,j} \geq 0$, и $p_i(x)$ са различни неразложими полиноми над $\mathbb{F}_q$ със старши коефициент 1, то тогава

$$\mathrm{lcm}(f_1(x), \ldots, f_t(x)) = p_1(x)^{\max\{e_{1,1}, \ldots, e_{t,1}\}} \ldots p_n(x)^{\{e_{1,n}, \ldots, e_{t,n}\}}.$$

Лема 6.1. Нека $f(x), f_1(x), f_2(x), \ldots, f_t(x)$ са полиноми над $\mathbb{F}_q$. Ако $f(x)$ се дели на всеки от полиномите $f_i(x)$, $i = 1, \ldots, t$, то $f(x)$ се дели и на $\mathrm{lcm}(f_1(x), f_2(x), \ldots, f_t(x))$.

Доказателство. Да пологим $g(x) = \mathrm{lcm}(f_1(x), f_2(x), \ldots, f_t(x))$. Съществуват полиному $u(x)$ и $r(x)$ над $\mathbb{F}_q$, $\deg(r(x)) < \deg(g(x))$, за които

$$f(x) = u(x)g(x) + r(x).$$

Оттук $r(x) = f(x) - u(x)g(x)$ и, следователно, $r(x)$ се дели на всички полиноми $f_i(x)$. Тъй като $g(x)$ е от минимална степен, $r(x) = 0$. □

Пример 6.2. Полиномът $f(x) = x^{15} - 1$ на д $\mathbb{F}_2$ се делина $f_1(x) = 1 + x + x^2 \in \mathbb{F}_2[x]$, $f_2(x) = 1 + x + x^4 \in \mathbb{F}_2[x]$ и $f_3(x) = (1 + x + x^2)(1 + x^3 + x^4) \in \mathbb{F}_2[x]$. Следователно $f(x)$ се дели и на $\mathrm{lcm}(f_1(x), f_2(x), f_3(x)) = (1 + x + x^2)(1 + x + x^4)(1 + x^3 + x^4)$. □

Пример 6.3. Да фиксираме примитивен елемент α на $\mathbb{F}_{q^M}$ и да означим с $M^{(i)}(x)$ минималния полином на α^i над $\mathbb{F}_q$. Всеки корен β на $M^{(i)}(x)$ е елемент на $\mathbb{F}_{q^m}$ и следователно $\beta^{q^m-1} - 1 = 0$, т.е. $x - \beta$ е делител на $x^{q^m-1} - 1$. Съгласно Теорема ??? полиномите $M^{(i)}(x)$ нямат кратни корени. Така съгласно Лема 6.1 за всяко подмножество I на $\mathbb{Z}_{q^m-1}$ най-малкото общо кратно $\mathrm{lcm}(M^{(i)}(x))_{i\in I}$ е делител на $x^{q^m-1} - 1$. □

В горния пример е описано как да намираме някои от делителите на $x^{q^m-1}-1$. Идеята е да изберем тези делители за пораждащи полиноми на кодове с дължина $q^m - 1$ над $\mathbb{F}_q$.

Дефиниция 6.4. Нека α е примитивен елемент на $\mathbb{F}_{q^m}$ и нека $M^{(i)}$ е минималния полином на α^i над $\mathbb{F}_q$. *(Примитивен) БЧХ-код* над $\mathbb{F}_q$ с дължина $n = q^m - 1$ и конструктивно разстояние δ наричаме цикличния код над $\mathbb{F}_q$, породен от полинома $g(x) := \mathrm{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$ за някое цяло число a. Казваме, че кодът е *БЧХ-код в тесен смисъл*, ако $a = 1$.

Пример 6.5. (i) Нека α е примитивен елемент на $\mathbb{F}_{2^m}$. Тогава БЧХ-код в тесен смисъл с конструктивно разстояние 2 е цикличния код, породен от $M^{(1)}(x)$. Оказва се, че това е кодът на Хеминг. (Да се докаже!)

(ii) Нека $\alpha \in \mathbb{F}_8$ е корен ан $1 + x + x^3 \in \mathbb{F}_2[x]$. Очевидно α е примитивен елемент на $\mathbb{F}_8$ и $M^{(1)}(x) = M^{(2)}(x) = 1 + x + x^3$. Следователно БЧХ-кодът в тесен смисъл с дължина 7, породен от $\mathrm{lcm}(M^{(1)}(x), M^{(2)}(x)) = 1 + x + x^3$ е $[7, 4]$. Това е двоичният $[7, 4, 3]$-код на Хеминг.

(iii) Нека α е същият като в (ii). Двоичният БЧХ-код с дължина 7, породен от

$$\mathrm{lcm}(M^{(0)}(x), M^{(1)}(x), M^{(2)}(x)) = \mathrm{lcm}(1 + x, 1 + x + x^3) = (1 + x)(1 + x + x^3)$$

е цикличен $[7, 3]$-код. Лесно се проверява, че това е ортогоналният на кода на Хеминг от (ii).

Пример 6.6. Нека β е корен на $1 + x + x^2 \in \mathbb{F}_2[x]$. Тогава $\mathbb{F}_2(\beta) = \mathbb{F}_4$. нека α е корен на $\beta + x + x^2 \in \mathbb{F}_4[x]$. Тогава α е примитивен елемент на $\mathbb{F}_{16}$. Да разгледаме БЧХ-код в тесен смисъл над $\mathbb{F}_4$ с дължина 15 и конструктивно разстояние 4. Пораждащият полином на този код е

$$g(x) = \mathrm{lcm}(M^{(1)}(x), M^{(2)}(x), M^{(3)}(x)) = 1 + \beta x + \beta x^2 + x^3 + x^4 + \beta^2 x^5 + x^6.$$

□

6.1.2 Параметри на БЧХ-кодове

Дължината на БЧХ-кодовете е по дефиниция $n = q^m - 1$. Сега ще разгледаме въпроса с размерността на БЧХ-кодовете.

Теорема 6.7. *(i)* Размерността на един q-ичен БЧХ-код с дължина q^m-1, породен от $g(x) := \mathrm{lcm}(M^{(a)}(X), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x)\}$ е наезависим от избора на примитивния елемент α.

(ii) Размерността на q-ичен БЧХ-код с дължина q^m-1 и конструктивно разстояние δ е поне $q^m-1-m(\delta-1)$.

Доказателство. (i) НекаC_i е циклотомичният клас на q по модул q^m-1, съдържащ i. да положим $S = \cup_{i=a}^{a+\delta-2} C_i$. Очевидно имаме

$$g(x) = \mathrm{lcm}\left(\prod_{i\in C_a}(x-\alpha^i), \prod_{i\in C_{a+1}}(x-\alpha^i), \ldots, \prod_{i\in C_{a+\delta-2}}(x-\alpha^i)\right) = \prod_{i\in S}(x-\alpha^i).$$

Следователно размерността е равна на $q^m-1-\deg(g(x)) = q^m-1-|S|$. Разулатът следва от това, че S не зависи от избора на α.

(ii) Съгласно (i) размерността k изпълнява

$$\begin{aligned} k &= q^m-1-|S| \\ &= q^m-1-\left|\cup i = a^{a+\delta-2}C_i\right| \\ &\geq q^m-1-\sum_{i=a}^{a+\delta-2}|C_i| \\ &\geq q^m-1-\sum_{i=a}^{a+\delta-2} m \\ &= q^m-1-m(\delta-1), \end{aligned}$$

което завъвършва доказателството. □

Горната теорема показва, че да определим размерността на q-ичен БЧХ-код сд ължина q^m-1, породен от $g(x) = \mathrm{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$, е достатъчно да проверим мощността на $\cup_{i=a}^{a+\delta-2} C_i$, където C_i е циклотомичният клас на q по модул q^m-1, съдържащ i.

Пример 6.8. (i) да разгледаме следните циклотомични класове на 2 по модул 15,

$$C_2 = \{1, 2.4, 8\},\ C_3 = \{3, 6, 12, 9\}.$$

Размерността на двоичния БЧХ-код с дължина 15 и конструктивно разстояние 3, породен от $g(x) = \mathrm{lcm}(M^{(2)}(x), M^{(3)}(x))$ е

$$15-|c_2 \cup C_3| = 15-8 = 7.$$

Тук се достига границата от Теорема 6.7.

(ii) да разгледаме следните циклотомични класове на 3 по модул 26

$$C_1 = C_3 = \{1, 3, 9\}, C_2\{= 2, 6, 18\}, C_4 = \{4, 12, 10\}.$$

размерността на троичния БЧХ-код с дължина 26 и конструктивно разстояние 5, породен от

$$g(x) := \mathrm{lcm}(M^{(1)}(x), M^{(2)}(X), M^{(3)}(x), M^{(4)}(x))$$

е

$$26 - |C_1 \cup C_2 \cup C_3 \cup C_4| = 26 - 9 = 17.$$

В този случай размерността е строго по-голяма от долната граница от Теорема 6.7. □

Пример 6.9. (i) За $t \geq 1$ целите числа t и $2t$ принадлежат на един и същи съседен циклотомичен клас на 2 по модул $2^m - 1$. Това е еквивалентно на факта, че $M^{(t)}(x) = M^{(2t)}(x)$. Следователно,

$$\mathrm{lcm}(M^{(1)}(x), \ldots, M^{(2t-1)}(x)) = \mathrm{lcm}(M^{(1)}(x), \ldots, M^{(2t)}(x)),$$

т.е. двоичният БЧХ-код в тесен смисъл с дължина $2^m - 1$ с конструктивно разстояние $2t + 1$ съвпада с двоичния БЧХ-код в тесен смисъл с дължина $2^m - 1$ с конструктивно разстояние $2t$.

таблицата по-долу съдържа размерностите на БЧХ-кодовете в тесен смисъл с дължина $2^m - 1$, $3 \leq m \leq 6$, с конструктивно разстояние $2t + 1$. Да отбележим отново, че размерността на БЧХ-код в тесен смисъл не зависи от избора на примитивен елемент.

n	k	t	n	k	t
7	4	1	63	51	2
15	11	1	63	45	3
15	7	2	63	39	4
15	5	3	63	36	5
31	26	1	63	30	6
31	21	2	63	24	7
31	16	3	63	18	10
31	11	5	63	16	11
31	6	7	63	10	13
63	57	1	63	7	15

(ii) Нека α е корен на $1 + x + x^4 \in \mathbb{F}_2[x]$; тогава α е примитивен елемент на $\mathbb{F}_{16}$. Да пресметнем минималните полиномите

$$\begin{aligned}
M(0)(x) &= 1 + x, \\
M^{(1)}(x) = M^{(2)}(x) = M^{(4)}(x) = M^{(8)}(x) &= 1 + x + x^4, \\
M^{(3)}(x) = M^{(6)}(x) = M^{(12)}(x) = M^{(9)}(x) &= 1 + x + x^2 + x^3 + x^4, \\
M^{(5)}(x) = M^{(10)}(x) &= 1 + x + x^2, \\
M^{(7)}(x) = M^{(14)}(x) = M^{(13)}(x) = M^{(11)}(x) &= 1 + x^3 + x^4.
\end{aligned}$$

Пораждащите полиноми на БЧХ-кодовете в тесен смисъл с дължина 15 са предсатавени по-долу.

n	k	t	Generator polynomial
15	11	1	$1+x+x^4$
15	7	2	$(1+x+x^4)(1+x+x^2+x^3+x^4)$
15	5	3	$(1+x+x^4)(1+x+x^2+x^3+x^4)(1+x+x^2)$

В Пример 6.8(ii) е показано, че границата от Теорема 6.7 може да се подобри в някои случаи. Следващата теорема дава достатъчни условия за достигане на долната граница от Теорема 6.7.

Теорема 6.10. Един q-ичен БЧХ-код в тесен смисъл с дължина 15 и конструктивно разстояние δ има размерност точно $q^m-1-m(\delta-1)$, ако $q\neq 2$ и $\gcd(q^m-1,e)=1$ за всяко $1\leq e\leq \delta-1$.

Доказателство. От доказателството на Теорема 6.7 знаем, че размерността на такъв БЧХ-код е равна на

$$q^m-1-\left|\cup_{i=1}^{\delta-1}C_i\right|,$$

където C_i са циклотомичните класове на q по модул q^m-1, съдържащи i. Следователно, достатъчно е да дпкажем, че $|C_i|=m$ за всички $1\leq i\leq \delta-1$, както и че C_i и C_j не се пресичат за всички $1\leq i<j\leq \delta-1$.

За всяко цяло $1\leq t\leq m-1$ и за всяко цяло $1\leq i\leq \delta-1$ е изпълнено $i\not\equiv q^t i \pmod{q^m-1}$. В противен случай бихме имали $(q^t-1)i\equiv 0 \pmod{q^m-1}$ и $(q^t-1)\equiv 0 \pmod{q^m-1}$, тъй като $\gcd(i,q^m-1)=1$. Това е невъзможно и, следователно, $|C_i|=m$ за всяко $1\leq i\leq \delta-1$.

Сега за всички цели числа $1\leq i<j\leq \delta-1$ и всяко $s\geq 0$ имаме $j\not\equiv q^s i \pmod{q^m-1}$. В противен случай, $j-i\equiv (q^s-1)i \pmod{q^m-1}$. Оттук следва $j-i\equiv 0 \pmod{q-1}$, което е противоречие с условието $\gcd(j-i,q^m-1)=1$. Следователно C_i и C_j не се пресичат. □

Пример 6.11. Да разгледаме БЧХ-код в тесен смисъл над $\mathbb{F}_4$ с дължина 63 и конструктивно разстояние 3. Размерността му е $63-3(3-1)=57$. □

Както вече видяхме, БЧХ-кодовете в тесен смисъл с конструктивно разстояние $2t$ са същите като БЧХ-кодовете в тесен смисъл с конструктивно разстояние $2t-1$. Следователно достатъчно е да разглеждаме БЧХ-кодове в тесен смисъл с нечетно конструктивно разстояние.

Теорема 6.12. Един двоичен БЧХ-код в тесен смисъл с дължина $n=2^m-1$ и конструктивно разстояние $\delta=2t+1$ има размерност поне $n-m(\delta-1)/2$.

Доказателство. Тъй като циклотомичните класове C_i и C_{2i} съвпадат, то за размерността k имаме

$$\begin{aligned}
k &= 2^m-1-\left|\cup_{i=1}^{2t}C_i\right| \\
&= 2^m-1-\left|\cup_{i=1}^{t}C_i\right| \\
&\geq 2^m-1-\sum_{i=1}^{t}|C_{2i-1}| \\
&\geq 2^m-1-tm \\
&= 2^m-1-m(\delta-1)/2.
\end{aligned}$$

□

Пример 6.13. Един БЧХ-код в тесен смисъл с дължина 63 и конструктивно разстояние 5 има размерност точно $51 = 63 - 6(5-1)/2$. От друга страна БЧХ-код с дължина 31 и конструктивно разстояние $\delta = 11$ има размерност 11, която е по-голяма от $31 - 5(11-1)/2$. □

Сега ще се спрем на задачата за определяне на минималното разстояние на БЧХ-кодове.

Лема 6.14. Нека C е q-ичен цикличен код с дължина n и пораждащ полином $g(x)$. Нека $\alpha_1, \ldots, \alpha_r$ се всички корени на $g(x)$ и да предположим, че той няма кратни корени. Елементът $c(x)$ на $\mathbb{F}_q[x]/(x^n-1)$ е кодова дума от C тогава и само тогава, когато $c(\alpha_i) = 0$ за $i = 1, \ldots, r$.

Доказателство. Ако $c(x)$ е кодова дума в C, то съществува полином $f(x)$, за който $c(x) = g(x)f(x)$. Сега имаме $c(\alpha_i) = g(\alpha_i)f(\alpha_i) = 0$ за всички $i = 1, \ldots, r$.

Обратно, нека $c(\alpha_i) = 0$ за $i = 1, \ldots, r$. Тогава $c(x)$ се дели на $g(x)$, тъй като $g(x)$ няма кратни корени. Това означава, че $c(x)$ е кодова дума на C. □

Пример 6.15. Да разгледаме двоичния $[7, 4]$-код на Хемингс пораждащ полином $g(x) = 1 + x + x^3$. Тъй като всички елементи на $\mathbb{F}_8 \setminus \{0, 1\}$ са корени на $c(x) = 1 + x + x^2 + x^3 + x^4 + x^5 + x^6 = (x^7 - 1)/(x - 1)$, то и всички корени на $g(x)$ са корени на $c(x)$. Следователно, 1111111 е кодова дума. □

Следващата теоремаизяснява понятието конструктивно разстояние на БЧХ-код.

Теорема 6.16. Един БЧХ-код с конструктивно разстояние δ има минимално разстояние поне δ.

Доказателство. Нека α е примитивен елемент на $\mathbb{F}_{q^m}$ и нека C е БЧХ-кодът, породен от $g(x) := \operatorname{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$. Ясно е, че елементите $\alpha^a, \alpha^{a+1}, \ldots, \alpha^{a+\delta-2}$ са корени на $g(x)$.

Да допуснем, че минималното разстояние d на C е по-малко от δ. Тогава съществува ненулева кодова дума $c(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}$, за която $w_{\mathrm{Ham}}(c(x)) = D < \delta$. Съгласно Лема 6.14, $c(\alpha^i) = 0$ за $i = a, a+1, \ldots, a+\delta-2$, т.е.

$$\begin{pmatrix} 1 & \alpha^a & (\alpha^a)^2 & \ldots & (\alpha^a)^{n-1} \\ 1 & \alpha^{a+1} & (\alpha^{a+1})^2 & \ldots & (\alpha^{a+1})^{n-1} \\ 1 & \alpha^{a+2} & (\alpha^{a+2})^2 & \ldots & (\alpha^{a+2})^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{a+\delta-2} & (\alpha^{a+\delta-2})^2 & \ldots & (\alpha^{a+\delta-2})^{n-1} \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{pmatrix} = \mathbf{0}. \tag{6.1}$$

Нека носителят на $c(x)$ (множеството от ненулеви координатни позиции) Е $R = \{i_1, \ldots, i_d\}$, т.е. $c_j \neq 0$ точно когато $j \in R$. Тогава от (**??**) получаваме

$$\begin{pmatrix} (\alpha^a)^{i_1} & (\alpha^a)^{i_2} & (\alpha^a)^{i_3} & \ldots & (\alpha^a)^{i_d} \\ (\alpha^{a+1})^{i_1} & (\alpha^{a+1})^{i_2} & (\alpha^{a+1})^{i_3} & \ldots & (\alpha^{a+1})^{i_d} \\ (\alpha^{a+2})^{i_1} & (\alpha^{a+2})^{i_2} & (\alpha^{a+2})^{i_3} & \ldots & (\alpha^{a+2})^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\alpha^{a+\delta-2})^{i_1} & (\alpha^{a+\delta-2})^{i_2} & (\alpha^{a+\delta-2})^{i_3} & \ldots & (\alpha^{a+\delta-2})^{i_d} \end{pmatrix} \begin{pmatrix} c_{i_1} \\ c_{i_2} \\ c_{i_3} \\ \vdots \\ c_{i_d} \end{pmatrix} = \mathbf{0}. \tag{6.2}$$

Тъй като $d \leq \delta - 1$, избирайки първите d уравнения от (**??**), получаваме следната система от линейни уравнения по отношение на неизвестните c_{i_j}:

$$\begin{pmatrix} (\alpha^a)^{i_1} & (\alpha^a)^{i_2} & (\alpha^a)^{i_3} & \dots & (\alpha^a)^{i_d} \\ (\alpha^{a+1})^{i_1} & (\alpha^{a+1})^{i_2} & (\alpha^{a+1})^{i_3} & \dots & (\alpha^{a+1})^{i_d} \\ (\alpha^{a+2})^{i_1} & (\alpha^{a+2})^{i_2} & (\alpha^{a+2})^{i_3} & \dots & (\alpha^{a+2})^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\alpha^{a+d-2})^{i_1} & (\alpha^{a+d-2})^{i_2} & (\alpha^{a+d-2})^{i_3} & \dots & (\alpha^{a+d-2})^{i_d} \end{pmatrix} \begin{pmatrix} c_{i_1} \\ c_{i_2} \\ c_{i_3} \\ \vdots \\ c_{i_d} \end{pmatrix} = \mathbf{0}. \tag{6.3}$$

Детерминантата D на матрицата от коефициентите на горната система е равна на

$$\begin{aligned} D &= \prod_{j=1}^{d} (\alpha^a)^{i_j} \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \alpha^{i_3} & \dots & \alpha^{i_d} \\ (\alpha^2)^{i_1} & (\alpha^2)^{i_2} & (\alpha^2)^{i_3} & \dots & (\alpha^2)^{i_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\alpha^{d-1})^{i_1} & (\alpha^{d-1})^{i_2} & (\alpha^{d-1})^{i_3} & \dots & (\alpha^{d-1})^{i_d} \end{vmatrix} \\ &= \prod_{j=1}^{d} (\alpha^a)^{i_j} \prod_{k>l} (\alpha^{i_k} - \alpha^{i_l}) \neq 0. \end{aligned}$$

Това означава, че хомогенната система (**??**) има единствено решение $c_{i_1}, \dots, c_{i_d}) = (0, \dots, 0)$, което противоречи на избора на $c(x)$. □

Пример 6.17. (i) Нека α е корен на $1 + x + x^3 \in \mathbb{F}_2[x]$ и нека C е двоичен БЧХ-кода с дължина 7 и конструктивно разстояние 4, породен от

$$g(x) = \operatorname{lcm}(M^{(0)}(x), M^{(1)}(x), M^{(2)}(x))) = 1 + x^2 + x^3 + x^4.$$

Тогава $d(c) \leq w_{\text{Ham}}(g(x)) = 4$. От друга страна, съгласно Теорема 6.16 $d(c) \geq 4$. Следователно, $d(C) = 4$.

(ii) Нека α е корен на $1 + x + x^4 \in \mathbb{F}_2[x]$. Очевидно α е примитивен елемент на $\mathbb{F}_{16}$. да разгледаме БЧХ-код в тесен смисъл с дължина 15 и конструктивно разстояние 7. Пораждащият полином на този код е

$$\begin{aligned} g(x) &= \operatorname{lcm}(M^{(1)}(x), M^{(2)}(x), \dots, M^{(6)}(x)) \\ &= M^{(1)}(x) M^{(3)}(x) M^{(5)}(x) \\ &= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}. \end{aligned}$$

Следователно, $d(C) \leq w_{\text{Ham}}(g(x)) = 7$, а от друга страна, съгласно Теорема 6.16 $d(C) \geq 7$. Следователно $d(C) = 7$. □

Пример 6.18. Нека α е примитивен елемент на $\mathbb{F}_{2^m}$ и нека $M^{(1)}(x)$ е минималния полином на α над $\mathbb{F}_2$. Да разгледаме двоичния БЧХ-код в тесен смисъл C с дължина $n = 2^m - 1$и конструктивно разстояние 3, породен от

$$g(x) = \operatorname{lcm}(M^{(1)}(x), M^{(2)}(x)) = M^{(1)}(x).$$

Тогава по Теорема 6.16 $d(C) \geq 3$ и C е двоичният код на Хеминг. □

6.1.3 Декодиране на БЧХ-кодове

6.2 Кодове на Рид-Соломон

Най-важният клас БЧХ-кодове е класът на т.нар. *кодове на Рид-Соломон*. Те са въведени от И. С. Рид и Г. Соломон, независимо от Хоквингем, Боуз и Рей-Чоудхури.

Да разгледаме q-ичен БЧХ-код C с дължина q^m-1, породен от полинома $g(x) := \mathrm{lcm}(M^{(a)}(x), M^{(a+1)}(x), \ldots, M^{(a+\delta-2)}(x))$, където $M^{(i)}(x)$ е минималният полином над $\mathbb{F}_q$ на α^i за избран примитивен елемент α. Ако $m=1$, получаваме q-ичен БЧХ-код с дължина $q-1$. В тои случай α е примитивен елемент на $\mathbb{F}_q$ и минималният полином на α^i над $\mathbb{F}_q$ е $x-\alpha^i$. Така за $\delta \leq q-1$ пораждащият полином приема вида

$$\begin{aligned} g(x) &= \mathrm{lcm}(x-\alpha^a, x-\alpha^{a+1}, \ldots, x-\alpha^{a+\delta-2}) \\ &= (x-\alpha^a)(x-\alpha^{a+1})\ldots(x-\alpha^{a+\delta-2}), \end{aligned}$$

тъй като елементите $\alpha^a, \alpha^{a+1}, \ldots, \alpha^{a+\delta-2}$ са два по два различни.

Дефиниция 6.19. БЧХ-кода над $\mathbb{F}_q$ с дължина $q-1$, породен от полинома

$$g(x) = (x-\alpha^{a+1})(x-\alpha^{a+2})\ldots(x-\alpha^{a+\delta-1})$$

със $a \geq 0$ и $2 \leq \delta \leq q-1$ и където α е примитивен елемент на $\mathbb{F}_q$ ще наричаме *q-ичен код на Рид-Соломон* или *RS*-код.

Няма да разглеждаме двоични RS-кодове, тъй като в този случай дължината е $q-1=1$.

Пример 6.20. Да разгледаме RS-код над $\mathbb{F}_7$ с дължина 6 и пораждащ полином $g(x) := (x-3)(x-3^2)(x-3^3) = 6+x+3x^2+x^3$. Това е $[6,3]$-код над $\mathbb{F}_7$. Една пораждаща матрица получаваме от $g(x)$:

$$G = \begin{pmatrix} 6 & 1 & 3 & 1 & 0 & 0 \\ 0 & 6 & 1 & 3 & 1 & 0 \\ 0 & 0 & 6 & 1 & 3 & 1 \end{pmatrix}.$$

Проверочна матрица получаваме от $h(x) = (x^6-1)/g(x) = 1+x+4x^2+x^3$. От получената проверочна матрица можем да проверим, че минималното разстояние е 4. така построеният RS-код е $[7,3,4]$ MDS-код над $\mathbb{F}_7$.

(ii) Да разгледаме RS-кода над $\mathbb{F}_8$ с дължина 7 и пораждащ полином $g(x) = (x-\alpha)(x-\alpha^2) = 1+\alpha+(\alpha^2+\alpha)x+x^2$, където α е корен на $1+x+x^3 \in \mathbb{F}_8$. Това е $[7,5]$-код над $\mathbb{F}_8$. От $g(x)$ можем да получим пораждаща матрица за този код:

$$G = \begin{pmatrix} \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 & 0 & 0 \\ 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 & 0 \\ 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 & 0 \\ 0 & 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 & 0 \\ 0 & 0 & 0 & 0 & \alpha+1 & \alpha^2+\alpha & 1 \end{pmatrix}.$$

Проверочната матрица

$$H = \begin{pmatrix} 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 & 0 \\ 0 & 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 \end{pmatrix}$$

се получава от

$$h(x) = (x^7 - 1)/g(x) = \alpha^4 + (1+\alpha^4)x + (1+\alpha^4)x^2 + x^3 + \alpha^4 x^4 + x^5.$$

От тази проверочна матрица можем да проверим, че минималното разстояние е 3. Следователно този код е $[7, 5, 3]$ MDS-код над $\mathbb{F}_8$. □

Теорема 6.21. Кодовете на Рид-Соломон са *MDS*-кодове, т.е. q ичният код на Рид-Соломон с дължина $q-1$, породен от $g(x) = \prod_{i=a+1}^{a+\delta-1}(x - \alpha^i)$ е цикличен код с параметри $[q-1, q-\delta, \delta]$ за всяко $2 \leq \delta \leq q-1$.

Доказателство. Тъй като степента на $g(x)$ е $\delta - 1$, то размерността на кода е точно $k := q - 1 - (\delta - 1) = q - \delta$. Съгласно Теорема 6.16 минмалното разстояние е поне δ. От друга страна минмалното разстояние е най-много $(q-1)+1-k = \delta$ съгласно гарницата на Сингълтън. Оттук следва искания резултат. □

Пример 6.22. Нека α е корен на $1 + x + x^4 \in \mathbb{F}_2[x]$. Тогава α е примитивен елемент на $\mathbb{F}_{16}$. Да разгледаме RS-кода над $\mathbb{F}_{16}$, породен от $g(x) = (x - \alpha^3)(x - \alpha * 4)(x - \alpha^5)(x - \alpha^6)$. От Теорема **??** получаваме, че минималното разстояние е 5, а кодът е с параметри $[15, 11, 5]$.

Теорема 6.23. Нека C е q-ичен *RS*-код, породен от полинома $g(x) = \prod_{i=1}^{\delta-1}(x - \alpha^i)$, където $2 \leq \delta \leq q = 1$. Тогава разширеният код $\overline{C}$ е също *MDS*-код.

Доказателство. Нека C е цикличен $[q-1, q-\delta, \delta]$-код. Трябва да покажем, че $\overline{C}$ е $[q, q-\delta, \delta+1]$-код. Нека $c(x) = \sum_{i=0}^{q-2} c_i x^i$ е ненулева дума от C. Достатъчно е да покажем, че теглото на $\overline{\boldsymbol{c}} = c_0, c_1, \ldots, c_{q-2}, -\sum_{i=0}^{q-2} c_i)$ е поне $\delta + 1$. Нека $c(x) = f(x)g(x)$ за някакъв полином $f(x) \in \mathbb{F}_q[x]/(x^{q-1} - 1)$.

Случай 1: $f(1) \neq 0$. Ясно е, че $g(1) \neq 0$. Следователно $c(1) = \sum_{i=0}^{q-2} c_i \neq 0$. Тогава теглото на Хеминг на $\overline{\boldsymbol{c}}$ е равно на $w_{\mathrm{Ham}}(c(x)) + 1$, което е поне $d(C) + 1 = \delta + 1$.

Случай 2: $f(1) = 0$, т.е. $x-1$ е линеен множител на $f(x)$. Да положим $f(x) = (x-1)u(x)$. Тогава $c(x) = u(x)(x-1)g(x) = u(x)\prod_{i=0}^{\delta-1}(x - \alpha^i)$ е също кодова дума от БЧХ-код с конструктивно разстояние $\delta + 1$, породен от $\prod_{i=0}^{\delta-1}(x - \alpha^i)$. Следователно, теглото на Хеминг на $c(x)$ е поне $\delta + 1$ (Теорема 6.16). Така теглото на $\overline{\boldsymbol{c}}$ е поне $\delta + 1$. □

Пример 6.24. (i) да разгледаме $[6, 3, 4]$-кода над $\mathbb{F}_7$ от Пример 6.20. Матрицата

$$\begin{pmatrix} 1 & 4 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

е проверочна матрица на разширения код. Кои да е четири стълба на тази матрица са линейно независими и следователно минималното разстояние 5. Така тя е проверочна матрица на $[7, 3, 5]$ MDS-код.

(ii) Да разгледаме RS-кода над $\mathbb{F}_8$ с параметри $[7,5,3]$ от Пример 6.20(ii). Матрицата

$$\begin{pmatrix} 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 & 0 & 0 \\ 0 & 1 & \alpha^4 & 1 & 1+\alpha^4 & 1+\alpha^4 & \alpha^4 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

е проверочна матрица на разширения код. така той има минимлано разстояние 4 и е $[8,5,4]$ MDS-код. □

Кодовете на Рид-Соломон са MDS-кодове и като такива имат добри параметри. за съжаление RS-кодовете не са двоични, докто практические приложения често изискват точно двоични кодове. Възможно е да използваме конкатенация на кодове за получаване на добри двоични кодове от RS-кодове над някакво разширение на $\mathbb{F}_2$.

Нека C е $[n,k]$-код на Рид-Соломон над $\mathbb{F}_{2^m}$, където $n = 2^m - 1$. използваме конкатенационната техника от Теорема 4.19, използвайки тривиалния код $\mathbb{F}_2^m$. Нека $\alpha_1, \ldots, \alpha_m$ е базис на $\mathbb{F}_{2^m}$ над $\mathbb{F}_2$ и да разгледаме изображението $\phi : \mathbb{F}_{2^m} \to \mathbb{F}_2^m$, зададено с

$$u + 1\alpha_1 + u_2\alpha_2 + \ldots + u_m\alpha_m \to (u_1, u_2, \ldots, u_m).$$

Тогава от Теорема 4.19 получаваме следния резултат.

Теорема 6.25. Нека C е $[n,k]$ *-RS* код над $\mathbb{F}_{2^m}$, където $n = 2^m - 1$. Тогава

$$\phi^*(C) := \{(\phi(c_0), \phi(c_1), \ldots, \phi(c_{n-1}) \mid (c_0, c_1, \ldots, c_{n-1}) \in C\}$$

е двоичен $[mn, mk]$-код с минимално разстояние поне $n - k = 1$.

Пример 6.26. да разгледаме RS-кода C, породен от

$$g(x) = \prod_{i=1}^{6}(x - \alpha^i) = \sum_{i=1}^{6} x^i,$$

където α е корен на $1 + x + x^3$. Следователно,

$$C = \{a(1,1,1,1,1,1,1) \mid a \in \mathbb{F}_8\}$$

е тривиалният $[7,1,7]$ MDS-код (код с повторение). Тогава кодът $\phi^*(C)$ е двоичен $[21,3,7]$-код, породен от

$$100100\ldots100,\ \ 010010\ldots010,\ \ 001001\ldots001.$$

□

При зададен RS-код C кодът $\phi^*(C)$ не може да поправя много случайни грешки, тъй като минималното резстояние не е голямо. Но този код може да се използва много по-успешно за поправяне на пакети от грешки.

Теорема 6.27. Нека C е $[2^m - 1, k]$-код на Рид-Соломон над $\mathbb{F}_{2^m}$. Тогава кодът $\phi^*(C)$ може да поправя $m\lfloor (n-k)/2 \rfloor - m + 1$ пакетни грешки, където $n = 2^m - 1$ е дължината на кода.

Доказателство. Да положим $l = m\lfloor (n-k)/2 \rfloor - m + 1$. Съгласно Теорема **??** е достатъчно да покажем ,че всички пакетни грешки с дължина l или по-малко лежат в различни съседни класове.

Нека $\boldsymbol{e}_1, \boldsymbol{e}_2 \in \mathbb{F}_2^{mn}$ са две пакетни грешки с дължина l или по-малко, които лежат в един и същи съседен клас на $\phi^*(C)$. нека $\boldsymbol{c}_i$ е прообразът на $\boldsymbol{e}_i$ при зиображението ϕ^*, т.е. $\phi^*(\boldsymbol{c}_i) = \boldsymbol{e}_i$ за $i = 1, 2$. Тогава е ясно, че

$$\begin{aligned} w_{\mathrm{Ham}}(\boldsymbol{c}_i) &= \lceil \frac{l-1}{m} \rceil + 1 \\ &= \lfloor \frac{n-k}{2} \rfloor \\ &= \lfloor \frac{d(C)-1}{2} \rfloor \ \text{(тъй като } C \text{ е MDS-код)} \end{aligned}$$

за $i = 1, 2$ и $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ са в един и същи съседен клас на C. Тогава $\boldsymbol{c}_1 = \boldsymbol{c}_2$, което означава, че $\boldsymbol{e}_1 = \boldsymbol{e}_2$, тъй като ϕ^* е инективно. □

Пример 6.28. За $[7, 3, 5]$-RS кода над $\mathbb{F}_8$, кодът $\phi^*(C)$ е линеен, двоичен $[21, 9]$-код. Той поправя

$$l = 3\lfloor \frac{7-3}{2} \rfloor - 3 + 1 = 4$$

пакетни грешки. □

6.3 Квадратично-остатъчни кодове

Нека $p > 2$ е просто число и нека g е примитивен елемент в $\mathbb{F}_p$. Един ненулев елемент $r \in \mathbb{F}_p$ се нарича *квадратичен остатък по модул* p, ако $r = g^{2i}$ за някакво цяло число i; в противен случай r се нарича *квадратичен неостатък по модул* p. ясно е, че r е квадратичен неостатък тогава и само тогава, когато $r = g^{2j-1}$ за някакво цяло число j.

Пример 6.29. (i) Да разгледаме крайното поле $\mathbb{F}_7$. Лесно се проверява, че 3 е примитвен елемент във $\mathbb{F}_7$. ненулевите квадратични остатъци по модул 7 са $\{3^{2i} \mid i = 0, 1, \ldots\} = \{1, 2, 4\}$, а квадратичните неостатъци са $\{3^{2i-1} \mid i = 1, 2, \ldots\} = \{3, 6, 5\}$.

(ii) В полето $\mathbb{F}_{11}$ 2 е примитивен елемент. Ненулевите квадратични остатъци по модул 11 са $\{2^{2i} \mid i = 0, 1, \ldots\} = \{1, 4, 5, 9, 3\}$, а квадратичните неостатъци са $\{2^{2i-1} \mid i = 1, 2, \ldots\} = \{2, 8, 10, 7, 6\}$.

(iii) Лесно се проверява, че в полето $\mathbb{F}_{23}$ 5 е примитивен елемент. така ненулевите квадратични остатъци по модул 23 са $\{5^{2i} \mid i = 0, 1, \ldots\} = \{1, 2, 4, 8, 16, 9, 18, 13, 3, 6, 12\}$, а квадратичните неостатъци са $\{5^{2i-1} \mid i = 1, 2, \ldots\} = \{5, 10, 20, 17, 11, 22, 21, 19, 15, 7, 14\}$. □

Теорема 6.30. Един ненулев елемент r във $\mathbb{F}_p$ е ненулев квадратичен остатък по модул p тогава и само тогава, когато $r \equiv a^2 \pmod{p}$ за някое $a \in \mathbb{F}_p^*$. По-специално квадратичните остатъци са независими от избора на примитивен елемент.

Теорема 6.31. Нека p е нечетно просто число. да означим с $\mathcal{Q}_p$ и $\mathcal{N}_p$ съответно множествата на ненулевите квадратични остатъци и неостатъци по модул p. В сила са твърденията:

(i) Произведението на два квадратични остатъка по модул p е квадратичен остатък по модул p.

(ii) Произведението на два квадратични неостатъка по модул p е квадратичен неостатък по модул p.

(iii) Произведението на ненулев квадратичен остатък по модул p и квадратичен неостатък по модул p е квадратичен неостатък по модул p.

(iv) Съществуват точно $(p-1)/2$ ненулеви квадратични остатъци по модул p и $(p-1)/2$ квадратични неостатъци по модул p. Следователно $\mathbb{F}_p = \{0\} \cup \mathcal{Q}_p \cup \mathcal{N}_p$.

(v) Ако $\alpha \in \mathcal{Q}_p$ и $\beta \in \mathcal{N}_p$, то е изпълнено

$$\begin{aligned} \alpha\mathcal{Q}_p &= \{\alpha r \mid r \in \mathcal{Q}_p\} = \mathcal{Q}_p, \\ \beta\mathcal{Q}_p &= \{\beta r \mid r \in \mathcal{Q}_p\} = \mathcal{N}_p, \\ \alpha\mathcal{N}_p &= \{\alpha n \mid n \in \mathcal{N}_p\} = \mathcal{N}_p, \\ \beta\mathcal{N}_p &= \{\beta n \mid n \in \mathcal{N}_p\} = \mathcal{Q}_p. \end{aligned}$$

Пример 6.32. да разгледаме $\mathbb{F}_{11}$. Множеството на ненулевите квадратични остатъци по модул 11 е $\mathcal{Q}_{11} = 1, 4, 5, 9, 3\}$, това на квадратичните неостатъци по модул 11 е $\mathcal{N}_{11} = \{2, 8, 10, 7, 6\}$. Очевидно $|\mathcal{Q}_{11}| = |\mathcal{N}_{11}| = 5$. По-нататък избирайки $4 \in \mathcal{Q}_{11}$ и $2 \in \mathcal{N}_{11}$ получаваме

$$\begin{aligned} 4\mathcal{Q}_{11} &= \{4, 5, 9, 3, 1\} = \mathcal{Q}_{11} \\ 2\mathcal{N}_{11} &= \{2, 8, 10, 7, 6\} = \mathcal{N}_{11} \\ 4\mathcal{Q}_{11} &= \{8, 10, 7, 6, 2\} = \mathcal{N}_{11} \\ 2\mathcal{N}_{11} &= \{4, 5, 9, 3, 1\} = \mathcal{Q}_{11} \end{aligned}$$

□

Да изберем просто число l, $l \neq p$, което е квадратичен остатък по модул p. да изберем цяло число $m \geq 1$, за което $l^m - 1$ се дели на p. Нека θ е примитивен елемент на $\mathbb{F}_{l^m}$ и да положим $\alpha = \theta^{(l^m-1)/p}$. Тогава редът на α е p; т.е. имаме $1 = \alpha^0, \alpha = \alpha^1, \alpha^2, \ldots, \alpha^{p-1}$ са два по два различни и $x^p - 1 = \prod_{i=0}^{p-1}(x - \alpha^i)$.

Да разгледаме полиномите

$$g_{\mathcal{Q}}(x) = \prod_{r \in \mathcal{Q}_p} (x - \alpha^r), \quad g_{\mathcal{N}}(x) = \prod_{n \in \mathcal{N}_p} (x - \alpha^n). \tag{6.4}$$

От Теорема 6.31(iv) следва, че

$$x^p - 1 = (x-1) g_{\mathcal{Q}}(x) g_{\mathcal{N}}(x).$$

Лема 6.33. Коефициентите на полиномите $g_{\mathcal{Q}}(x)$ и $g_{\mathcal{N}}(x)$ са във $\mathbb{F}_l$.

Доказателство. Нека $g_{\mathcal{Q}}(x) = a_0 + a_1 x + \ldots + a_k x^k$, където $a_i \in \mathbb{F}_{l^m}$, $k = (p-1)/2$. Имаме

$$\begin{aligned} a_0^l + a_1^l x + \ldots + a_k^l x^k &= \prod_{r \in \mathcal{Q}_p} (x - \alpha^{lr}) \\ &= \prod_{j \in l\mathcal{Q}_p} (x - \alpha^j) \\ &= \prod_{j \in \mathcal{Q}_p} (x - \alpha^j) \\ &= g_{\mathcal{Q}}(x). \end{aligned}$$

Тук използваме факта, че $l\mathcal{Q} = \mathcal{Q}$. Следователно $a_i = a_i^l$, $0 \leq i \leq m$, т.е. елементите a_i са от $\mathbb{F}_l$ и $g_{\mathcal{Q}}(x) \in \mathbb{F}_l$. Същия аргумент можем да използваме за полинома $g_{\mathcal{N}}(x)$. □

Пример 6.34. (i) Нека $p = 7$ и $l = 2$. Нека α е корен на $1 + x + x^3 \in \mathbb{F}_2[x]$. Полиномите, дефинирани в (6.4) са

$$\begin{aligned} g_{\mathcal{Q}}(x) &= \prod_{r \in \mathcal{Q}_7} (x - \alpha^r) \\ &= (x - \alpha)(x - \alpha^2)(x - \alpha^4) \\ &= 1 + x + x^3 \end{aligned}$$

и

$$\begin{aligned} g_{\mathcal{N}}(x) &= \prod_{n \in \mathcal{N}_7} (x - \alpha^n) \\ &= (x - \alpha^3)(x - \alpha^6)(x - \alpha^5) \\ &= 1 + x^2 + x^3 \end{aligned}$$

Непосредствено се проверява, че $x^7 - 1 = (x-1) g_{\mathcal{Q}}(x) g_{\mathcal{N}}(x)$.

(ii) нека $p = 11$ и $l = 3$. Нека θ е корен на $1 + 2x + x^5 \in \mathbb{F}_3[x]$. Тогава θ е примитивен елемент на $\mathbb{F}_{3^5}$ и редът на $\alpha := \theta^{22}$ е 11. Сега полиномите, дефинирани в (6.4) са

$$\begin{aligned} g_{\mathcal{Q}}(x) &= \prod_{r \in \mathcal{Q}_{11}} (x - \alpha^r) \\ &= (x - \alpha)(x - \alpha^3)(x - \alpha^9)(x - \alpha^5)(x - \alpha^4) \\ &= 2 + x^2 + 2x^3 + x^4 + x^5 \end{aligned}$$

и

$$\begin{aligned} g_{\mathcal{N}}(x) &= \prod_{n \in \mathcal{N}_{11}} (x - \alpha^n) \\ &= (x - \alpha^2)(x - \alpha^6)(x - \alpha^7)(x - \alpha^{10})(x - \alpha^8) \\ &= 2 + 2x + x^2 + 2x^3 + x^5. \end{aligned}$$

Непосредствено се проверява, че $x^{11} - 1 = (x-1)((x^5 + x^4 + 2x^3 + x^2 + 2)((x^5 + 2x^3 + x^2 + 2x + 2)$.

(iii) нека $p = 23$ и $l = 2$. Нека θ е корен на $1 + x + x^3 + x^5 + x^{11} \in \mathbb{F}_2[x]$. Тогава θ е примитивен елемент на $\mathbb{F}_{2^{11}}$ и редът на $\alpha := \theta^{89}$ е 23. Двата полинома, дефинирани в (6.4) са

$$\begin{aligned} g_{\mathcal{Q}}(x) &= \prod_{r\in\mathcal{Q}_{23}} (x-\alpha^r) \\ &= (x-\alpha)(x-\alpha^2)(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16}) \\ &\quad \times(x-\alpha^9)(x-\alpha^{18})(x-\alpha^{13})(x-\alpha^3)(x-\alpha^6)(x-\alpha^{12}) \\ &= 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11} \end{aligned}$$

и

$$\begin{aligned} g_{\mathcal{N}}(x) &= \prod_{n\in\mathcal{N}_{23}} (x-\alpha^n) \\ &= (x-\alpha^5)(x-\alpha^{10})(x-\alpha^{20})(x-\alpha^{17})(x-\alpha^{11}) \\ &\quad \times(x-\alpha^{22})(x-\alpha^{21})(x-\alpha^{19})(x-\alpha^{15})(x-\alpha^7)(x-\alpha^{14}) \\ &= 2 + 2x + x^2 + 2x^3 + x^5. \end{aligned}$$

□

Дефиниция 6.35. Нека p и l са различни прости числа и нека l е квадратичен остатък по модул p. Да изберем цяло число $m \geq 1$, за което p дели $l^m - 1$. Нека θ е примитивен елемент на $\mathbb{F}_{l^m}$ и да положим $\alpha = \theta^{(l^m-1)/p}$. Делителите на $x^p - 1$

$$g_{\mathcal{Q}}(x) := \prod_{r\in\mathcal{Q}_p} (x-\alpha^r) \text{ и } g_{\mathcal{N}}(x) := \prod_{n\in\mathcal{N}} (x-\alpha^n)$$

са дефинирани над $\mathbb{F}_l$. Цикличните кодове $C_{\mathcal{Q}} = \langle g_{\mathcal{Q}}(x)$ и $C_{\mathcal{N}} = \langle g_{\mathcal{N}}(x)\rangle$ с дължина p се наричат *квадратично-остатъчни кодове* или QR-кодове.

Пример 6.36. (i)
(ii)
(iii)

В горните три примера кодовете $C_{\mathcal{Q}}$ и $C_{\mathcal{N}}$ са еквивалентни. Оказва се, че това е общ факт, който ще докажем. започваме с една спомагателна лема.

Лема 6.37. Нека $m, n > 1$ са цели числа и нека $(m, n) = 1$. Тогава изображението

$$\chi_m : \mathbb{F}_q[x]/(x^n - 1) \to \mathbb{F}_q[x]/(x^n - 1), \;\; a(x) \mapsto a(x^m)$$

е пермутация на $\mathbb{F}_q^n$, ако идентифицираме $\mathbb{F}_q^n$ с $\mathbb{F}_q[x]/(x^n - 1)$ чрзе изображението

$$\pi : (f_0, f_1, \ldots, f_{n-1}) \mapsto \sum_{i=0}^{n-1} f_i x^i.$$

Доказателство.

Пример 6.38.

Теорема 6.39. Квадратично-остатъчните кодове $C_{\mathcal{Q}}$ и $C_{\mathcal{N}}$ над $\mathbb{F}_l$ са еквивалентни.

Доказателство.

Теорема 6.40. *(i)* Нека p е нечетно просто число и нека r е цяло,з а което $\gcd(r,p)=1$. Числото r е квадратичен остатък по модул p тогава и само тогава, когато $r^{(p-1)/2} \equiv 1 \pmod{p}$.

(ii) Нека p е нечетно просто число. Числото 2 е квадратичен остатък по модул p, ако p е отвида $p = 8m \pm 1$ и е квадратичен неостатък, ако p е от вида $p = 8m \pm 3$.

Следствие 6.41. Двоични квадратично-остатъчни кодове с дължина p съществуват тогава и само тогава, когато p е просто число от вида $p = 8m \pm 1$.

Пример 6.42. В таблицата по-долу представяме параметрите на първите няколко двоични квадратично-остатъчни кодове.

Дължина	размерност	Минмално разстояние
7	4	3
17	9	5
23	12	7
31	16	7
41	21	9
47	24	11
71	36	11
73	37	13
79	40	15
89	45	17

6.4 Задачи

1.

Глава 7

Кодове на Гоппа

През 1970 г. Витали Гоппа описва нощ интересен клас от линейни кодове, които получават името *кодове на Гоппа*. Този клас включва вече разгледаните БЧХ-кодове в тесен смисъл. Оказва се, че кодовете на Гоппа са най-интересния подклас от алтернантни кодове, въведени от Хелгерт през 1874 г.

7.1 Обобщени кодове на Рид-Соломон

В раздел 6.2 въведохме кодовете на Рид-Соломон като специален клас БЧХ-кодове. Код на Рид-Соломон над $\mathbb{F}_q$ дефинирахме като БЧХ-код над $\mathbb{F}_q$ с дължина $q-1$, породен от

$$g(x) = (x-\alpha^a)(x-\alpha^{a+1})\dots(x-\alpha^{a+\delta-2}),$$

където $a \geq 1$ и $q-1 \geq \delta \geq 2$ и α е примитивен елемент на $\mathbb{F}_q$. Този код е MDS-код с параметри $[q-1, q-\delta, \delta]$. Сега да разгледаме кодове на Рид-Соломон в тесен смисъл, т.е. $a = 1$. В този случаи9 съществува алтернативно описание на RS-кдовете, което ще се окаже удобно.

Теорема 7.1. Нека α е примитивен елемент на $\mathbb{F}_q$ и нека $q-1 \geq \delta \geq 2$. Кодът на Рид-Соломон в тесен смисъл над $\mathbb{F}_q$, имащ пораждащ полином

$$g(x) = (x-\alpha)(x-\alpha^2)\dots(x-\alpha^{\delta-1})$$

е равен на

$$\{(f(1), f(\alpha), f(\alpha^2), \dots, f(\alpha^{q-2})) \mid f(x) \in \mathbb{F}_q[x], \deg f(x) < q-\delta\}. \tag{7.1}$$

Доказателство. Лесно се проверява, че множеството, зададено с (**??**) е векторно пространство над $\mathbb{F}_q$. най-напред ще покажем, че то се съдържа в RS-кода породен от $g(x)$.

Кодовата дума $\boldsymbol{c} = (f(1), f(\alpha), f(\alpha^2), \dots, f(\alpha^{q-2}))$ съответства на полинома $c(x) = \sum_{i=0}^{q-2} f(\alpha^i)x^i \in \mathbb{F}_q[x]/(x^n-1)$. Трябва да покажем, че $c(x)$ се дели на $g(x)$, с други думи, че

$$c(\alpha) = c(\alpha^2) = \dots = c(\alpha^{\delta-1} = 0.$$

Да отбележим, че за $1 \leq k \leq q-2$ имаме $\sum_{i=0}^{q-2} \alpha^{ik} = ((\alpha^k)^{q-1} - 1)?(q-1) = 0$. Нека $f(x) = \sum_{j=0}^{q-\delta-1} f_j x^j$. Тогава за всяко $l = 1, \ldots, \delta - 1$ имаме

$$c^{\alpha^l} = \sum_{i=0}^{q-2} f(\alpha^i)(\alpha^l)^i = \sum_{i=0}^{q-2} \left(\sum_{j=0}^{q-\delta-1} f_j \alpha^{ij} \right) \alpha^{il} = \sum_{j=0}^{q-\delta-1} f_j \left(\alpha^{i(j+l)} \right) = 0.$$

тъй като $1 \leq j + l \leq q - 2$.

Изображението $f \mapsto (f(1), f(\alpha), \ldots, f(\alpha^{q-2}))$ от полиномите в $\mathbb{F}_q[x]$ от степен $< q - \delta$ в множеството, зададено с (**??**), е инективно. Ако $f(x)$ е в ядрото на това изображение, то той трябва да има $q-1 > \deg f(x)$ нули, откъдето е тъждествено 0. Това изображение е и сюрективно и следователно и изоморфизъм на векторни пространства. Размерността на векторното пространство от (**??**) щ $q - \delta$ и съвпада с размерносста на RS-кода, породен от $g(x)$. Това доказва теоремата. □

Следствието по-долу дава друга експлицитна пораждаща матрица за кдовете на Рид-Соломон в тесен смисъл.

Следствие 7.2. Нека α е примитивен елемент на $\mathbb{F}_q$ и нека $q - 1 \geq \delta \geq 2$. Тогава матрицата

$$\begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \alpha^{q-2} \\ 1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(q-2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{q-\delta-1} & \alpha^{2(q-\delta-1)} & \ldots & \alpha^{(q-\delta-1)(q-2)} \end{pmatrix}$$

е пораждаща матрица за кода на Рид-Соломон с пораждащ полином

$$g(x) = (x - \alpha)(x - \alpha^2) \ldots (x - \alpha^{\delta-1}).$$

Сега ще обобщим дефиницията от Теорема 7.1 за да получим един по-общ клас от кодове, които също са MDS-кодове.

Дефиниция 7.3. Нека $n \leq q$. Нека $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$, където α_i, $i = 1, \ldots, n$, са различни елементи на $\mathbb{F}_q$. Нека $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$, където $v_i \in \mathbb{F}_q^*$ за всички $1 \leq i \leq n$. За всяко $k \leq n$ дефинираме *обобщения код на Рид-Соломон* $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ посредством

$$\{(v_1 f(\alpha_1), v_2 f(\alpha_2), \ldots, v_n f(\alpha_n)) \mid f(x) \in \mathbb{F}_q, \deg f(x) < k\}$$

Елементите $\alpha_1, \alpha_2, \ldots, \alpha_n$ наричаме *локатори на кода* $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$.

Теорема 7.4. Обобщения код на Рид-Соломон $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ има параметри $[n, k, n-k+1]$, т.е. това е *MDS*-код.

Доказателство. Очевидно $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ е с дължина n. Както в теорема 7.1 можем да покажем, че размерността на този код е k. Остава да докжем , че минималното разстояние е $n - k + 1$.

Ще преброим максималния брой нули в ненулева кодова дума. Да предположим, че $f(x)$ не е тъждествено нула. Тъй като $\deg f(x) < k$, полиномът може да има най-много $k-1$ нули. С други думи, кодовата дума

$$(v_1 f(\alpha_1), v_2 f(\alpha_2), \ldots, v_n f(\alpha_n))$$

има най-много $k-1$ нулеви координанти. Така теглото ѝ е поне $n-k+1$ и миналното разстояние d на $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ удовлетворя $d \geq n-k+1$. От границата на Сингълтън следва, че $d = n-k+1$, т.е. $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ е MDS-код. □

Забележка 7.5. В случая, когато $\boldsymbol{v} + (1, 1, \ldots, 10$ и $n < q-1$ конструираният обобщен код на Рид-Соломон се нарича често *съкратен код на Рид-Соломон*, тъй може да бъде получен от RS-код чрез съкращаване на подходящи координати.

Както и при кодовете на Рид-Соломон, ортогоналният на един обобщен код на Рид-Соломон е отново обобщен код на Рид-Соломон.

Теорема 7.6. Ортогоналният код на обобщения кдо на Рид-Соломон $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ над $\mathbb{F}_q$ с дължина n е $\mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \boldsymbol{v}')$ за някакъвектор $\boldsymbol{v}' \in (\mathbb{F}_q^*)^n$.

Доказателство. Нека най-напред $k = n-1$. Известно е, че ортогоналният на $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$, който е MDS-код, е отново MDS-код и така параметрите му са $[n, 1, n]$. Базисът му се състои от вектора $\boldsymbol{v}' = (v - 1', v_2', \ldots, v_n')$, където $v_i' \in \mathbb{F}_q^*$. така ортогоналният код е $\mathrm{GRS}_1(\boldsymbol{\alpha}, \boldsymbol{v}')$. по специално иамме, че за всички $f(x) \in \mathbb{F}_q[x]$ от степен $< n-1$

$$v_1 v_1' f(\alpha_1) + v_2 v_2' f(\alpha) + \ldots v_n v_n' f(\alpha_n) = 0, \tag{7.2}$$

където $\boldsymbol{v} + (v_1, v_2, \ldots, v_n)$.

За произволно k твърдим, че $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})^\perp = \mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \boldsymbol{v}')$. Произволна кодова дума от $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ има вида$(v_1 f(\alpha_1), v_2 f(\alpha_2) \ldots, v_n f(\alpha_n))$, където $f(x) \in \mathbb{F}_q[x]$, $\deg f(x) \leq k-1$. От друга страна една кодова дума от $\mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \boldsymbol{v}')$ има вида $(v_1' g(\alpha_1), v_2' g(\alpha_2) \ldots, v_n' g(\alpha_n))$, където $g(x) \in \mathbb{F}_q[x]$, $\deg f(x) \leq n-k-1$. Тъй като $\deg(f(x)g(x0) \leq n-2 < n-1$, имаме

$$(v_1 f(\alpha_1), v-2 f(\alpha_2) \ldots, v_n f(\alpha_n)) \cdot (v_1' g(\alpha_1), v_2' g(\alpha_2) \ldots, v_n' g(\alpha_n)) = 0$$

от(7.2).

Следователно, $\mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \boldsymbol{v}') \subseteq \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})^\perp$. Резултатът следва като сравним размерностите на двата кода. □

Следствие 7.7. Матрицата

$$\begin{pmatrix} v'_1 & v'_2 & \dots & v'_n \\ v'_1\alpha_1 & v'_2\alpha_2 & \dots & v'_n\alpha_n \\ v'_1\alpha_1^2 & v'_2\alpha_2^2 & \dots & v'_n\alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ v'_1\alpha_1^{n-k-1} & v'_2\alpha_2^{n-k-1} & \dots & v'_n\alpha_n^{n-k-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha_1 & \alpha_2 & \dots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \dots & \alpha_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \dots & \alpha_n^{n-k-1} \end{pmatrix} \begin{pmatrix} v'_1 & 0 & \dots & 0 \\ 0 & v'_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & v'_n \end{pmatrix}$$

е проверочна матрица за $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$.

7.2 Алтернантни кодове

Алтернантните кодове са широк клас от линейни кодове, включващи кодовете на Хеминг и БЧХ-кодовете. В този раздел ще считаме, че RS-кодовете са дефинирани по-общо над $\mathbb{F}_{q^m}$.

Дефиниция 7.8. *Алтернантният код* $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ над крайното поле $\mathbb{F}_q$ е ограничението над крайното поле $\mathbb{F}_q$ на обобщения код на Рид-Соломон $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ над $\mathbb{F}_{q^m}$.

С други думи алтернантният код $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ еограничението $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$.

Теорема 7.9. Алтернантният код $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ има параметри $[n, k', d]$, където $mk-(m-1)n \leq k$ и $d \geq n-k+1$.

Доказателство. Съгласно Теорема 7.4, кодът $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ има параметри $[n, k, n-k+1]$. Следователно $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ има дължина n и резмерността му k' удовлетворя $k' \leq k$. Сега резултатът следва от Теорема **??**. □

От дефиницията на алтернантни кодове следва, че $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ е множеството на думите $\{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{c}H^T = \boldsymbol{0}\}$, където H е матрицата от Следствие 7.7.

Всеки елемент $\beta \in \mathbb{F}_{q^m}$ може да бъде записан по единствен начин във вида $\beta = \sum_{i=0}^{m-1} \beta_i \alpha^i$, където α е примитивен елемент на $\mathbb{F}_{q^m}$ и $\beta_i \in \mathbb{F}_q$, $1 = 0, 1, \dots, m-1$. Следователно, ако заменим всеки елемент β на H с вектора-стълба$(\beta_0, \beta_1, \dots, \beta_{m-1})^T$, ще получим $(n-k)m \times n$ матрица $\overline{H}$ над $\mathbb{F}_q$, чрез която $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ се определя като

$$\{\boldsymbol{c} \in \mathbb{F}_q^n \mid \boldsymbol{c}\overline{H}^T = \boldsymbol{0}\}.$$

Матрицата $\overline{H}$ играе роля на проверочна матрица на $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$. Единствената разлика е, че редовете на $\overline{H}$ са линейно зависими.

Пример 7.10. (i) Нека $q = 2$, а $m \geq 3$ е цяло число. Нека α е примитивен елемент на $\mathbb{F}_{2^m}$. да положим

$$\boldsymbol{v}' = (1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2}).$$

За всяко $\boldsymbol{\alpha} = (\alpha, \ldots, \alpha_{2^m-1})$, където $\{\alpha_1, \ldots, \alpha_{2^m-1}\} = \mathbb{F}_{2^m}^*$, алтернантният код $\mathcal{A}_{2^m-2}(\boldsymbol{\alpha}, \boldsymbol{v}')$ е

$$\mathcal{A}_{2^m-2}(\boldsymbol{\alpha}, \boldsymbol{v}') = \{\boldsymbol{c} \in \mathbb{F}_2^{2^m-1} \mid \boldsymbol{c}(1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2})^T = \boldsymbol{0}\}.$$

Ясно, че ако $H + (1, \alpha, \alpha^2, \ldots, \alpha^{2^m-2})$, то $\overline{H}$ е $m \times (2^m - 1)$ матрица, чиито стълбове са всички ненулеви вектори от $\mathbb{F}_2^m$. Това е проверочната матрица надвоичния код на Хеминг $\mathrm{Ham}(m, 2)$ и така $\mathcal{A}_{2^m-2}(\boldsymbol{\alpha}, \boldsymbol{v}') = \mathrm{Ham}(m, 2)$.

(ii) За всяко q и m дефинирахме БЧХ код над $\mathbb{F}_q$ като множеството от всички вектори $\boldsymbol{c} \in \mathbb{F}_q^n$, удовлетворяващи $\boldsymbol{c}H'^T = \boldsymbol{0}$, където

$$\begin{aligned} H' &= \begin{pmatrix} 1 & \alpha^a & \alpha^{2a} & \ldots & \alpha^{a(n-1)} \\ 1 & \alpha^{a+1} & \alpha^{2(a+1)} & \ldots & \alpha^{(a+1)(n-1)} \\ 1 & \alpha^{a+2} & \alpha^{2(a+2)} & \ldots & \alpha^{(a+2)(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{a+\delta-2} & \alpha^{2(a+\delta-2)} & \ldots & \alpha^{(a+\delta-2)(n-1)} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{\delta-2} & \alpha^{2(\delta-2)} & \ldots & \alpha^{(\delta-2)(n-1)} \end{pmatrix} \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & \alpha^a & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha^{a(n-1)} \end{pmatrix} \end{aligned}$$

е във вида, представен в Следствие 7.7. Следователно БЧХ-кодовете са алтернантни кодове.

(iii) Нека $q = 2$ и $m = 3$. Нека α е примитивен елемент на $\mathbb{F}_8$, удовлетворяващ $\alpha^3 + \alpha + 1 = 0$. Да положим $\boldsymbol{v}' = (1, \ldots, 1)$ и $\boldsymbol{\alpha} = (\alpha, \alpha^2, \ldots, \alpha^6)$. Тогава $\mathcal{A}_3(\boldsymbol{\alpha}, \boldsymbol{v}') = \{\boldsymbol{c} \in \mathbb{F}_2^6 \mid \boldsymbol{c}H^T = \boldsymbol{0}\}$, където

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^2 & \alpha^4 & \alpha^6 & \alpha & \alpha^3 & \alpha^5 \end{pmatrix}.$$

Тогава

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

която има следната редуцирана горна трапецовидна форма

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

Следователно $\mathcal{A}_3(\boldsymbol{\alpha}, \boldsymbol{v}')$ има порждаща матрица

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

и е $[6, 2, 4]$-код. □

По долу е представено описание на ортогоналния на алтернантен код, което следва непосредствено от Теореми 4.27 и 7.6.

Теорема 7.11. Ортогоналният на алтернантния код $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ е кодът

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \boldsymbol{v}').$$

Следващата теорема демонстрира съществуването на алтернантни кодове с дадени параметри.

Теорема 7.12. Нека са дадени положителни цели числа n, h, δ, m. Съществува алтернантен код $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ над $\mathbb{F}_q$, който е код дефиниран над подполе за *RS*-код с параметри $[n, k', d]$ над $\mathbb{F}_{q^m}$, където $k' \geq h$ и $d \geq \delta$, винаги когато

$$\sum_{w-0}^{\delta-1} (q-1)^w \binom{n}{w} < (q^m - 1)^{\lfloor (n-h)/m \rfloor}. \tag{7.3}$$

Доказателство. За всеки вектор $\boldsymbol{c} \in \mathbb{F}_q^n$ нека

$$R(\boldsymbol{\alpha}, k, \boldsymbol{c}) = \{\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)^n \mid \boldsymbol{c} \in \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})\}.$$

Да запишем $\boldsymbol{c} = (c_1, \ldots, c_n)$ и $\boldsymbol{v} = (v_1, \ldots, v_n)$. Имаме $c_i = v_i f(\alpha_i)$, $1 \leq i \leq n$, където $f(x) \in \mathbb{F}_q[x]$ е от степен $< k$. За фиксирана думе $\boldsymbol{c}$ полиномът $f(x)$ се определя еднозначно щом изберем k стойности за v_i. Следователно

$$|R(\boldsymbol{\alpha}, k, \boldsymbol{c})| \leq (q^m - 1)^k.$$

Броят на векторите $\boldsymbol{c} \in \mathbb{F}_q^n$ с тегло $< \delta$ се задава с $\sum_{w=0}^{\delta-1} \binom{n}{w}(q - 10^w$ и така, взимайки $k = n - \lfloor (n-h)/m \rfloor$, получаваме

$$\left| \bigcup_{\substack{w_{\mathrm{Ham}}(\boldsymbol{c}) < \delta \\ \boldsymbol{c} \in \mathbb{F}_q^n}} R(\boldsymbol{\alpha}, k, \boldsymbol{c}) \right| \leq \left(\sum_{w=0}^{\delta-1} \binom{n}{w} (q-1)^w \right) (q^m - 1)^{n - \lfloor (n-h)/m \rfloor}.$$

Сега

$$|(\mathbb{F}_{q^m}^*)^n| = (q^m - 1)^n.$$

Следователно от условието (7.3) получваме, че

$$\bigcup_{\substack{w_{\mathrm{Ham}}(\boldsymbol{c}) < \delta \\ \boldsymbol{c} \in \mathbb{F}_q^n}} R(\boldsymbol{\alpha}, k, \boldsymbol{c})$$

е строго по-малко от $(\mathbb{F}_{q^m}^*)$, т.е. съществува вектор $\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)$, за който $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ не съдържа дума от $\mathbb{F}_q^n$ с тегло $< \delta$. Следователно алтернантният код $\mathcal{A}_k(\boldsymbol{\alpha}, \boldsymbol{v}')$ е с минимално разстояние $\geq \delta$. Тъй като $k = n - \lfloor (n-h)/m \rfloor \geq ((m-1)n+h)/m$, от Теорема 7.9 получаваме $k' \geq mk - (m-1)n \geq h$. □

7.3 Кодове на Гоппа

Литература

[1] R.C. Bose, D.K. Ray-Chaudhuri, On a class of error correcting binary group codes, *Inform. and Control* **3**(1960), 68–79.

[2] R.C. Bose, D.K. Ray-Chaudhuri, Further results on error correcting binary group codes, *Inform. and Control* **3**(1960), 279–290.

[3] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley and Sons, 1991.

[4] E.N. Gilbert, A comparisonof sigmaling alphabets, *Bell System Tech. J.* **31**(1952), 504–522.

[5] D.C. Gorenstein, M. Zierler, A class of error-correcting codes in p^m symbols, *J. SIAM* **9**(1961), 207–214.

[6] W. Heise, P. Quattrocchi, Informations- und Codierungstheorie, Springer Verlag, Berlin, 3rd Edition, 1995.

[7] C.R.P. Hartmann, K.K. Tzeng, Generalizations of the BCH bound, *Inform. and Control* **20**(1972), 489–498.

[8] A. Hocquenghem, Codes correcteurs d'erreurs, *Chiffres*(Paris) **2**(1959), 147–156.

[9] D.A. Huffman, A method for the construction of minimum redundancy codes, *Proc. IRE* **40**(1952), 1098–1101.

[10] J.H. van Lint, R. M. Wilson, On the minimum distance of cyclic codes, *IEEE Trans. Inf. Theory* **IT-32**(1986), 23–40.

[11] E. Prange, Cyclic error-correcting codes in two symbols, AFCRC-TN-57 **103**, September (1957).

[12] C. Roos, A new lower bound on the minimum distance of the cyclic code, *IEEE Trans. Inf. Theory* **IT-29**(1983, 330–332.