# Klaidas taisančių kodų teorija

Paskaitų konspektai

Gintaras Skersys
VU MIF

2025 m. ruduo
2025 m. rugsėjo 8 d.

# Turinys

# Įžanga

Šioje mokymo priemonėje pateikiami klaidas taisančių kodų teorijos pagrindai. Autorius šį dalyką dėsto jau daugiau kaip dvidešimt metų ir reguliariai papildo, pataiso šią mokymo priemonę, atsižvelgdamas į dėstymo metu sukauptą patirtį.

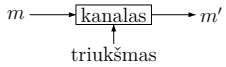
Ženklu „□“ žymėsime įrodymų ir pavyzdžių pabaigą.

Norėčiau padėkoti šio dalyko klausytojams Justui Kranauskui ir Andriui Unguriui už pradinio mokymo priemonės varianto surinkimą.

# 1 skyrius
# Pagrindinės sąvokos

## 1.1 Įvadas

Panagrinėkime informacijos perdavimo klausimą. Norime kažkam perduoti pranešimą $m$. Perduodant pranešimą, galimas jo iškraipymas. Tai galima pavaizduoti grafiškai šitaip:




Pranešimas $m$ perduodamas nepatikimu ryšio kanalu, t. y. kanale jį gali iškraipyti triukšmas. Iš kanalo išėjęs pranešimas $m'$ gali skirtis nuo $m$. Ką daryti, kad iškraipymo tikimybė būtų kuo mažesnė?

**1.1.1 pavyzdys.** Panagrinėkime kelis kanalų pavyzdžius. Visi šie ryšio kanalai yra *nepatikimi*, t. y. gali iškraipyti[1] informaciją.

1. Telefono linija (perduodama informacija gali būti iškraipyta dėl sąveikos su kitomis linijomis, triukšmo gali pridėti ir pačios linijos įranga).

2. Radijo ryšiu kosminis zondas siunčia Marso nuotraukas į Žemę (giliojo kosmoso antenos, besiklausančios zondo siunčiamų silpnų signalų, pagauna ir foninį spinduliavimą iš įvairių Žemės ir kosmoso šaltinių).

3. Ląstelių dalijimasis — motininės ląstelės DNR perduoda informaciją dukterinės ląstelės DNR (dėl radiacijos ir kitų veiksnių perduota informacija gali skirtis nuo pradinės – ląstelės gali mutuoti).

4. Kietasis diskas — informacija į jį užrašoma, o po kurio laiko nuskaitoma (kietojo disko įrašymo ir nuskaitymo įrenginys įrašo vieną bitą (t. y. nuliuką arba vienetuką) įmagnetindamas mažytį disko plotelį viena iš dviejų galimų krypčių, ir paskui gali jį blogai nuskaityti, pavyzdžiui, dėl to, kad tas plotelis spontaniškai pakeitė įmagnetinimo kryptį, arba terminis triukšmas sutrukdė nuskaitymo įrenginiui teisingai nuskaityti, arba įrašymo įrenginys negalėjo teisingai įmagnetinti plotelio dėl sąveikos su kaimyniniais bitais ir t. t.).

Paskutinis pavyzdys rodo, kad „kanalas“ nebūtinai reiškia, kad informacija perkeliama iš vienos vietos į kitą. Mes užrašome informaciją į kietąjį diską ir ją nuskaitome paprastai toje pačioje vietoje, bet skirtingu laiku. O per tą laiką ji galėjo būti iškraipyta. □

Matome, kad bet kuriuo atveju, mums perduodant informaciją, yra tam tikra tikimybė, kad gauta informacija nebus identiška išsiųstai informacijai. Norėtųsi turėti tokį ryšio kanalą, kuriame informacijos iškraipymo tikimybė būtų lygi nuliui, arba bent jau tokia artima nuliui, kad praktikoje ją galėtume laikyti lygia nuliui. Ką daryti?

*Fizinis sprendimas* būtų bandyti pagerinti ryšio kanalo fizines charakteristikas, kad klaidų tikimybė sumažėtų.

[1] Čia turima omenyje, kad iškraipymai įvyksta ne dėl kieno nors piktos valios, o dėl gamtinių sąlygų, technikos netobulumo ir pan. atsiradusio kanalo triukšmo. Piktavalius iškraipymus nagrinėja *kriptografija*.

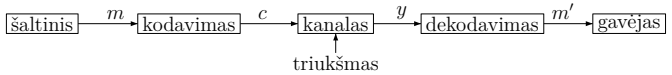


**1.1.2 pavyzdys.** Kietojo disko atveju, galima būtų pagerinti įrašymo ir nuskaitymo įrenginio darbo kokybę, pavyzdžiui,

- naudojant patikimesnius komponentus,
- išsiurbiant orą iš disko aplinkos, kad sūkurinės jėgos neveiktų nuskaitymo galvutės,
- naudojant didesnį plotelį kiekvieno bito atvaizdavimui,
- naudojant stipresnius signalus ar atšaldymą terminio triukšmo sumažinimui. □

Tačiau, pagerinus ryšio kanalo fizines charakteristikas, išauga perdavimo kaštai.

*Kodavimo teorijos siūlomas sprendimas* — priimame kanalą tokį, koks jis yra, bet, perduodami juo informaciją, naudojame tam tikrus metodus, padedančius aptikti ir ištaisyti kanale padarytas klaidas. Tam informacija prieš siunčiant į kanalą yra koduojama, o išėjusi iš kanalo — dekoduojama:




Detaliau panagrinėkime informacijos perdavimo etapus.

- Kodavimo metu prie pradinio pranešimo $m$ prijungiama papildoma informacija, leisianti aptikti ir ištaisyti tam tikrą skaičių kanale padarytų klaidų. Gaunamas užkoduotas pranešimas $c$, kuris yra didesnės apimties nei $m$.
- Pranešimas $c$ siunčiamas ryšio kanalu, kur galbūt yra kažkiek iškraipomas. Iš kanalo išeina $y$, kuris gali skirtis nuo $c$.
- Dekodavimo metu pranešime $y$, naudojantis kodavimo metu pridėta informacija, yra ištaisomos klaidos (jei tai įmanoma), bei yra gaunamas pranešimas $m'$, kuris bus lygus pradiniam pranešimui $m$, jei pranešime $y$ visos klaidos buvo ištaisytos.

Paprastai, naudojant tokią informacijos perdavimo sistemą su protingai parinktais kodavimo ir dekodavimo algoritmais, tikimybė, kad $m'$ skirsis nuo $m$, žymiai sumažėja (lyginant su atitinkama tikimybe, kai pranešimas siunčiamas be kodavimo), užtat išauga informacijos kiekis, siunčiamas kanalu.

Taigi, fiziniais būdais didinant kanalo patikimumą, jo kaina išauga, o naudojant kodavimą, vieninteliai kaštai yra papildomi skaičiavimai koduojant ir dekoduojant bei didesnis kanalo apkrovimas. Atkreipkite dėmesį į tai, kad, nors patikimo kanalo ir negalime turėti, tačiau tai nekliudo sukurti patikimą informacijos perdavimo sistemą, sudarytą iš trijų elementų: kodavimo procedūros, kanalo ir dekodavimo algoritmų.

*Informacijos teorija* nagrinėja tokių sistemų teorines ribas ir galimybes.

*Klaidas taisančių kodų teorija* (trumpiau dažnai vadinama *kodavimo teorija*), kurios pradmenys ir bus pateikti šiame kurse, kuria praktinius kodavimo ir dekodavimo būdus.

## 1.2 Paprasti klaidas aptinkančių ir taisančių kodų pavyzdžiai

Ką galima pridėti prie pradinio pranešimo, kad būtų galima aptikti ir ištaisyti kanale padarytas klaidas? Šiame skyriuje pateiksime kelis paprastus pavyzdžius. Griežti vartojamų sąvokų apibrėžimai (kas tai yra kodas ir t. t.) bus pateikti vėliau.

Tarkime, pradinis pranešimas $m$ yra *dvejetainis* (angl. *binary*) vektorius (t. y. *bitų* 0 ir 1 seka). Žymėsime $k$ jo ilgį (t. y. bitų skaičių jame). Paprastumo dėlei dvejetainį (o dažnai ir ne dvejetainį) vektorių rašysime be skliaustų ir kablelių, pavyzdžiui, vektorių $(1, 0, 1)$ užrašysime tiesiog 101.

### 1.2.1 Pakartojimo kodas

#### Kodavimas

Tarkime, $n \geqslant 1$ — sveikasis skaičius. Paprasčiausias kodavimo būdas — pakartoti kiekvieną siunčiamą ženklą (0 arba 1) $n$ kartų. Tiksliau, $k = 1$, pradinis pranešimas $m$ gali būti tik $m = 0$ arba $m = 1$ ($m$ yra ilgio $k = 1$ vektorius). Pradinis pranešimas $m = 0$ užkoduojamas vektoriumi $c = 00\cdots0$ (čia 0 pasikartoja $n$ kartų), o $m = 1$ užkoduojamas vektoriumi $c = 11\cdots1$ (čia 1 pasikartoja $n$ kartų). Toks kodas vadinamas *pakartojimo kodu* ir žymimas $R_n$.

**1.2.1 pavyzdys.** Tegu $n = 3$. Tada pakartojimo kodas $R_3$ $m = 0$ užkoduoja vektoriumi $c = 000$, o $m = 1$ — vektoriumi $c = 111$.

Pavyzdžiui, šaltinis siunčia bitų seką 1011. Ją skaidome į $k = 1$ ilgio pradinius pranešimus $m$, ir kiekvieną iš jų koduojame atskirai. Užkodavę gauname bitų seką 111 000 111 111. □

#### Dekodavimas

Dekoduojame daugumos principu: iš kanalo gautą $n$ ilgio pranešimą $y$ keičiame ženklu (bitu), kuris dažniausiai kartojasi tame pranešime. Norėdami dekoduoti bitų seką, ją skaidome į $n$ ilgio vektorius $y$, ir kiekvieną tokį vektorių dekoduojame atskirai.

**1.2.2 pavyzdys.** Tegu $n = 3$. Jei iš kanalo gavome pranešimą $y = 101$, tai dekoduojame $m' = 1$, nes pranešime $y$ vienetų yra daugiau, negu nulių. Jei $y = 001$, tai analogiškai $m' = 0$. Jei $y = 111$, tai $m' = 1$. □

#### Klaidų taisymas

**1.2.3 pavyzdys.** Tegu $n = 3$. Matome, kad jei kanalu siunčiamame pranešime padaryta viena klaida (pavyzdžiui, pradinis pranešimas buvo $m = 1$, užkoduotas ir į kanalą pasiųstas pranešimas $c = 111$, o iš kanalo išėjo $y = 101$), tai ji visada bus ištaisyta (nes likusiose dviejose pozicijose liko teisingi ženklai, ir jų bus daugiau, negu klaidingų, todėl dekoduosime teisingai), t. y. visada dekodavimo metu gautas $m'$ bus lygus $m$. Tačiau jei įvyks dvi klaidos, klaidingų ženklų bus daugiau, negu teisingų, ir dekoduosime klaidingai. □

Bendru atveju nesunku pastebėti, kad jei klaidų įvyko mažiau, nei pusėje visų pozicijų, tai toks dekodavimo algoritmas dekoduos teisingai, t. y. dekodavimo metu gautas $m'$ bus lygus pradiniam pranešimui $m$ (tokiu atveju sakysime, kad dekodavimo algoritmas ištaisė visas kanale padarytas



klaidas). Jei klaidų daugiau, nei pusė pozicijų, tai dekoduos klaidingai. Jei klaidingos lygiai pusė visų pozicijų, tai nežinosime, ar dekoduoti nuliu, ar vienetu, ir jei, pavyzdžiui, dekoduosime nuliu, galime dekoduoti klaidingai. Nesunkiai patikrinsite, kad bendru atveju dekoduosime *tikrai* teisingai (naudodami nurodytą dekodavimo algoritmą), jei iš kanalo gautame pranešime $y$ yra ne daugiau kaip $\left\lfloor \frac{n-1}{2} \right\rfloor$ klaidų, kur $\lfloor x \rfloor$ yra skaičiaus $x$ sveikoji dalis (didžiausias sveikasis skaičius, mažesnis už $x$ arba jam lygus). Todėl sakome, kad pakartojimo kodas $R_n$ taiso $\left\lfloor \frac{n-1}{2} \right\rfloor$ klaidų. Jei klaidų daugiau, tai galime dekoduoti klaidingai (bet galime ir teisingai).

**1.2.4 pavyzdys.**

- Kai $n = 3$, tikrai galime ištaisyti $\left\lfloor \frac{3-1}{2} \right\rfloor = \lfloor 1 \rfloor = 1$ klaidą.
- Kai $n = 4$, tikrai galime ištaisyti irgi $\left\lfloor \frac{4-1}{2} \right\rfloor = \lfloor 1{,}5 \rfloor = 1$ klaidą (jei įvyktų dvi klaidos, nulių ir vienetų gautume po lygiai, todėl negalėtume pasakyti, ar užkoduotas nulis, ar vienetas). □

#### Klaidų aptikimas

Šis kodas yra $n - 1$ klaidą aptinkantis kodas. Iš tikro, jei kanale padaryta ne daugiau kaip $n - 1$ klaida (bet bent viena klaida buvo padaryta), tai iš kanalo gausime pranešimą $y$, kuriame ne visi ženklai bus vienodi, iš ko ir nuspręsime, kad buvo klaidų. Bet jei klaidų buvo daugiau kaip $n - 1$, t. y. jų buvo $n$, tai gausime pranešimą, kur vėl visi ženklai bus vienodi, ir negalėsime nuspręsti, ar klaidų buvo.

**1.2.5 pavyzdys.** Tegu $n = 3$.

- Tarkime, iš kanalo gavome $y = 110$. Tada:
  - arba pradinis pranešimas buvo $m = 0$, užkoduotas ir į kanalą pasiųstas pranešimas buvo $c = 000$, ir siuntimo kanalu metu padarytos dvi klaidos (pirmojoje ir antrojoje pozicijose),
  - arba pradinis pranešimas buvo $m = 1$, užkoduotas ir į kanalą pasiųstas pranešimas buvo $c = 111$, ir siuntimo kanalu metu padaryta viena klaida (trečiojoje pozicijoje).

  Bet kuriuo atveju galime padaryti išvadą, kad kanale buvo padaryta klaidų. Sakome, kad kodas *aptiko* klaidas.
- Tarkime, iš kanalo gavome $y = 111$. Tada:
  - arba pradinis pranešimas buvo $m = 0$, užkoduotas ir į kanalą pasiųstas pranešimas buvo $c = 000$, ir siuntimo kanalu metu padarytos trys klaidos (visose trijose pozicijose),
  - arba pradinis pranešimas buvo $m = 1$, užkoduotas ir į kanalą pasiųstas pranešimas buvo $c = 111$, ir siuntimo kanalu metu klaidų padaryta nebuvo.

  Šiuo atveju jau negalime tvirtinti, kad kanale buvo padaryta klaidų (kaip ir negalime tvirtinti, jog jų padaryta nebuvo). □

#### Kodo koeficientas

*Kodo koeficientu* vadinsime pradinio ir persiunčiamo pranešimų ilgių santykį. Jis parodo, kuri į kanalą pasiųstų ženklų dalis yra naudinga informacija, o kuri tik pridėta klaidų aptikimui ir ištaisymui. Kuo jis didesnis, tuo geriau, nes tuo daugiau naudingos informacijos yra siunčiamame pranešime.

Pakartojimo kodo $R_n$ koeficientas yra $\frac{1}{n}$ (iš $n$ kanalu persiųstų ženklų tik vienas yra pradinio pranešimo ženklas).

#### Apibendrinimas

Pakartojimo kodo $R_n$ koeficientas yra mažas, tik $\frac{1}{n}$. Šis kodas nėra praktiškas. Bet klaidų jis ištaiso daug — beveik pusė ženklų gali būti klaidingi, vis tiek kodas dekoduos teisingai.

Klaidas taisančių kodų teorijos tikslas yra rasti tokius kodus, kad ir klaidų ištaisytų daug, ir kodo koeficientas būtų gana didelis.

### 1.2.2 Kontrolinio simbolio kodas

Jei kanale padaroma nedaug klaidų ir yra atgalinis ryšys su informacijos šaltiniu, t. y. yra galimybė paprašyti šaltinio pakartoti informacijos siuntimą, tai galbūt paprasčiau yra naudoti kodą, kuris klaidų nesugeba ištaisyti, bet gali greitai ir paprastai aptikti, kada žodis perduotas neteisingai. Aptikus tokią situaciją, galima paprašyti persiųsti žodį iš naujo. Kontrolinio simbolio kodas aptinka vieną klaidą su labai mažom sąnaudom.

#### Kodavimas

*Kontrolinio simbolio kodas* veikia taip. Pradinį pranešimą $m = m_1 m_2 \cdots m_k$ užkoduojame vektoriumi $c = m_1 m_2 \cdots m_k m_{k+1}$, kur *kontrolinis simbolis* $m_{k+1}$ prirašomas taip, kad vektoriuje $c$ būtų lyginis vienetų skaičius. Nesunku pastebėti, kad $m_{k+1}$ tenkina tokią formulę:

$$m_{k+1} = \left(\sum_{i=1}^{k} m_i\right) \bmod 2 = \begin{cases} 0, \text{ jei vienetų skaičius vektoriuje } m \text{ lyginis;} \\ 1, \text{ jei vienetų skaičius vektoriuje } m \text{ nelyginis.} \end{cases}$$

Prisiminkime, kad reiškinio $a \bmod q$ rezultatas yra liekana, gauta $a$ padalinus iš $q$. Pastebėsime, kad $-1 \bmod 2 = 1$ ir $2 \bmod 2 = 0$.

**1.2.6 pavyzdžiai.** 1. Jei $m = 1011$, tai $c = 10111$.

2. Jei $m = 1001$, tai $c = 10010$. □

Taip pat nesunku pastebėti, kad vektorius $c$ turi tenkinti lygybę

$$\sum_{i=1}^{k+1} m_i \equiv 0 \pmod 2.$$

Čia žymėjimas $a \equiv b \pmod q$ (skaitome „$a$ lygsta $b$ moduliu $q$“) reiškia, kad $a$ ir $b$ yra ekvivalentūs moduliu $q$, t. y. padalinę $a$ iš $q$ gausime tą pačią liekaną, kaip ir padalinę $b$ iš $q$. Pavyzdžiui, $-1 \equiv 1 \pmod 2$, $3 \equiv 1 \pmod 2$, $2 \equiv 0 \pmod 2$.



#### Dekodavimas

Dekodavimas vyksta taip. Patikriname gautame žodyje $y = y_1 y_2 \cdots y_k y_{k+1}$ vienetų skaičių.

- Jei vienetų skaičius lyginis, tai padarome išvadą, kad klaidų nėra, tai dekoduodami tiesiog atmetame paskutinę koordinatę, t. y. laikome, kad $m' = y_1 y_2 \cdots y_k$ ir yra pradinis pranešimas.
- Jei vienetų skaičius nelyginis, tada žinome, kad kanale buvo padaryta klaidų, todėl paprašome šaltinio perduoti žodį dar kartą.

#### Klaidų taisymas

Šis kodas klaidų netaiso.

#### Klaidų aptikimas

Jei kanale padaryta viena klaida, kontrolinio simbolio kodas visada ją aptiks. Iš tikrųjų, į kanalą siunčiami žodžiai tik su lyginiu vienetų skaičiumi. Įvykus vienai klaidai, vienetų skaičius žodyje pasidarys nelyginis, todėl bus galima padaryti išvadą, kad kanale įvyko klaidų.

Dviejų klaidų šis kodas neaptiks, nes įvykus dviems klaidoms vienetų skaičius žodyje vėl pasidarys lyginis, todėl nebus aišku, ar klaidų nebuvo visai, ar jų buvo lyginis skaičius.

Taigi, šis kodas yra 1 klaidą aptinkantis kodas. Kartu jis aptinka bet kokį nelyginį skaičių klaidų.

#### Kodo koeficientas

Kodo koeficientas $\frac{k}{k+1} = 1 - \frac{1}{k+1}$ yra labai aukštas, artimas vienetui, t. y. $k$ ženklų iš $k+1$ perduoto ženklo yra informacijos ženklai, tik 1 yra pridėtas klaidų aptikimui.

#### Apibendrinimas

Šis kodas klaidų netaiso, bet paprastai ir greitai aptinka nelyginį skaičių klaidų. Taip pat šio kodo koeficientas yra labai aukštas, t. y. jis beveik neapkrauna kanalo papildoma informacija, skirta klaidų aptikimui. Praktikoje jis dažniausiai naudojamas kartu su kitais klaidas taisančiais kodais, papildydamas jų galimybes.

Kontrolinio simbolio idėją galima panaudoti ne tik dvejetainiams kodams. Kituose dviejuose pavyzdžiuose nagrinėsime du tokius jau daugelį metų sėkmingai praktikoje taikomus kodus.

### 1.2.3 Knygų numeracijos sistema ISBN

Kiekviena šiuo metu leidžiama knyga turi unikalų ISBN (angl. *International Standard Book Number*) numerį, nurodantį šalių grupę, leidyklą ir knygos numerį. Iki 2007 metų ISBN numeris buvo sudarytas iš dešimties dešimtainių skaitmenų (vadinamas ISBN-10, ISO 2108 standartas, priimtas 1970 metais). Nuo 2007 metų buvo pereita jau prie 13 dešimtainių skaitmenų ISBN numerio (ISBN-13). Šiame poskyryje panagrinėsime ISBN-10 sudarymą.

**Kodavimas**

ISBN-10 numeris sudarytas iš devynių dešimtainių skaitmenų $a_1, \ldots, a_9$ bei dešimtojo kontrolinio skaitmens $a_{10}$. Kontrolinis simbolis $a_{10}$ pridedamas pagal tokią taisyklę:

$$a_{10} = \left( \sum_{i=1}^{9} i a_i \right) \bmod 11. \tag{1.1}$$

Jei gauname $a_{10} = 10$, tai kontrolinis simbolis $a_{10}$ žymimas ženklu $X$.

**1.2.7 pavyzdžiai.** 1. Viliaus Stakėno knygos „Informacijos kodavimas“, išleistos Vilniaus universiteto leidykloje 1996 metais, ISBN numeris yra 9986-19-183-1.

2. Harry Harrison knygos „Plieninė žiurkė keliauja pragaran“, išleistos leidyklos „Eridanas“ 1996 metais, ISBN numeris yra 9986-486-36-X.

3. Daugiau ISBN-10 pavyzdžių rasite pavartę nuo 1970 iki 2007 metų išleistas knygas. □

**1.2.8 užduotis.** Įrodykite, kad (1.1) lygybė yra ekvivalenti kiekvienai iš šių dviejų lygybių:

$$\sum_{i=1}^{10} i a_i \equiv 0 \pmod{11}, \tag{1.2}$$

$$\sum_{i=1}^{10} (11 - i) a_i \equiv 0 \pmod{11}. \tag{1.3}$$

**Dekodavimas**

Apskaičiuojamas kontrolinis simbolis $a_{10}$ pagal (1.1) formulę. Jei apskaičiuotasis sutampa su nurodytu ISBN numeryje, tai ISBN numeris galiojantis, o jei ne, tai ISBN numeryje yra klaidų.

ISBN numerį galima patikrinti ir naudojant (1.2) bei (1.3) lygybes.

**1.2.9 užduotis.** Paimkite bet kurią knygą su ISBN-10 numeriu ir patikrinkite, ar tikrai jos ISBN-10 numeris tenkina (1.1) lygybę.

**Klaidų taisymas**

Šis kodas klaidų netaiso.

**Klaidų aptikimas**

ISBN-10 kodas aptinka klaidas, kurios dažniausiai pasitaiko renkant skaičius:

- vieno skaitmens pakeitimas kitu (pavienė klaida),
- greta stovinčių skaitmenų sukeitimas vietomis (transpozicija).

Jei padaromos dvi klaidos, šis kodas jų gali ir neaptikti.

**1.2.10 užduotis.** 1. Įrodyti, kad (1.1) lygybė nebegalioja, jei padaroma pavienė klaida arba transpozicija.

2. Rasti tokį dviejų klaidų pavyzdį, kad (1.1) lygybė išliktų teisinga.



**Apibendrinimas**

ISBN-10 numeris aptinka dažniausias renkant skaičius pasitaikančias klaidas, todėl jis padėjo išvengti nemažai nesusipratimų, knygynams užsakinėjant knygas iš leidyklų. Deja, ir jis nepadeda, jei leidykla išleidžia knygą su negaliojančiu ISBN numeriu. O tokių atsitikimų irgi yra pasitaikę.

**1.2.11 užduotis.** Pasidomėkite, kaip sudaromas ISBN-13 numeris.

### 1.2.4 Lietuvos piliečio asmens kodas

Lietuvos piliečių asmens kodo struktūra yra tokia:

$$L\ Y_1Y_2M_1M_2D_1D_2\ X_1X_2X_3\ K,$$

kur:

- $L$ reiškia lytį ir gimimo šimtmetį:

  1 — vyras, gimęs XIX a.,
  2 — moteris, gimusi XIX a.,
  3 — vyras, gimęs XX a.,
  4 — moteris, gimusi XX a.,
  5 — vyras, gimęs XXI a.,
  6 — moteris, gimusi XXI a.

- $Y_1Y_2M_1M_2D_1D_2$ yra asmens gimimo data:
  - $Y_1Y_2$ — metai šimtmetyje,
  - $M_1M_2$ — mėnuo,
  - $D_1D_2$ — diena,
- $X_1X_2X_3$ — asmens, gimusio tą dieną, eilės numeris (priskiriamas Gyventojų registro),
- $K$ — kontrolinis skaičius.

Kontrolinis skaičius $K$ apskaičiuojamas, dauginant kiekvieną asmens kodo skaitmenį iš koeficiento ir sumuojant moduliu 11:

$$S = (L \cdot 1 + Y_1 \cdot 2 + Y_2 \cdot 3 + M_1 \cdot 4 + M_2 \cdot 5 + D_1 \cdot 6 + D_2 \cdot 7 + X_1 \cdot 8 + X_2 \cdot 9 + X_3 \cdot 1) \bmod 11.$$

Jei $S \neq 10$, tai $K = S$. Jei $S = 10$, tai skaičiuojama nauja suma su kitokiais koeficientais:

$$S' = (L \cdot 3 + Y_1 \cdot 4 + Y_2 \cdot 5 + M_1 \cdot 6 + M_2 \cdot 7 + D_1 \cdot 8 + D_2 \cdot 9 + X_1 \cdot 1 + X_2 \cdot 2 + X_3 \cdot 3) \bmod 11.$$

Jei $S' \neq 10$, tai $K = S'$. Jei $S' = 10$, tai $K = 0$.

**1.2.12 užduotis.** Patikrinkite, ar tikrai jūsų asmens kodas tenkina šias lygybes.

### 1.2.5 Lentelės kodas

Grįžkime vėl prie bitų sekų. Kontrolinio simbolio kodas apsaugo siunčiamą pranešimą, pridėdamas vieną kontrolinį simbolį. Jei jų pridėtume daugiau, galbūt pavyktų ne tik aptikti klaidas, bet jas ir ištaisyti? Šiame ir kitame poskyriuose panagrinėsime du pavyzdžius, kaip tai galima padaryti — lentelės kodą ir Hemingo kodą. Šie kodai yra vieną klaidą taisantys kodai.

#### Kodavimas

Pradinis pranešimas $m$ yra ilgio 4 vektorius. Surašome jį į $2 \times 2$ lentelę ir kiekvienai lentelės eilutei bei kiekvienam stulpeliui prirašome po vieną kontrolinį simbolį tokiu pačiu principu, kaip kontrolinio simbolio kodo atveju — kad vienetų skaičius kiekvienoje eilutėje ir kiekviename stulpelyje pasidarytų lyginis. Koduotą vektorių $c$ gausime eilutę po eilutės nuskaitę gautą lentelę.

**1.2.13 pavyzdys.** Pavyzdžiui, tegu $m = 1101$. Surašome jį į $2 \times 2$ lentelę:

$$\begin{array}{cc} 1 & 1 \\ 0 & 1 \end{array}$$

Dabar kiekvienai lentelės eilutei bei kiekvienam stulpeliui prirašome po vieną kontrolinį simbolį:

$$\begin{array}{cc|c} 1 & 1 & 0 \\ 0 & 1 & 1 \\ \hline 1 & 0 & \end{array}$$

Tada koduotas vektorius bus $c = 11001110$. □

#### Dekodavimas

Pažiūrėkime, koks turėtų būti dekodavimo algoritmas, kad visada ištaisytų pavienę klaidą.

Iš kanalo gautą vektorių $y$ surašome į lentelę. Kas atsitiks, jei jame bus viena klaida? Tai priklauso nuo to, kur ta klaida buvo padaryta.

- Jei klaida padaryta pozicijose, priklausančiose pradiniam pranešimui, tai kažkurioje eilutėje ir kažkuriame stulpelyje vienetų skaičius taps nelyginis. Radę tokią eilutę ir tokį stulpelį, ištaisome klaidą jų susikirtime.
- Jei klaida kuriame nors iš kontrolinių simbolių, tai tik vienoje eilutėje (ar stulpelyje) vienetų skaičius taps nelyginis. Sutikę tokią situaciją, ištaisome klaidą atitinkamame kontroliniame simbolyje.

**1.2.14 pavyzdžiai.** 1. Tarkime, klaida padaryta antrojoje vektoriaus iš paskutinio pavyzdžio $c = 11001110$ pozicijoje, t. y. iš kanalo išėjo vektorius $y = 10001110$. Surašome jį į lentelę ir patikriname vienetų skaičių kiekvienoje eilutėje ir kiekviename stulpelyje. Matome, kad pirmojoje eilutėje ir antrajame stulpelyje vienetų skaičius yra nelyginis (pažymėta rodyklėmis). Nusprendžiame, kad klaida įvyko jų susikirtime (apibrauktas):

$$\begin{array}{cc|c|l} 1 & \boxed{0} & 0 & \leftarrow \\ 0 & 1 & 1 & \\ \hline 1 & 0 & & \\ & \uparrow & & \end{array}$$



Ištaisę klaidą antrojoje pozicijoje, gauname vektorių $c' = 11001110$. Matome, kad jis sutampa su $c$, t. y. klaidą ištaisėme.

2. Tarkime, klaida padaryta šeštojoje vektoriaus iš paskutinio pavyzdžio $c = 11001110$ pozicijoje, t. y. iš kanalo išėjo vektorius $y = 11001010$. Surašome jį į lentelę ir patikriname vienetų skaičių kiekvienoje eilutėje ir kiekviename stulpelyje. Matome, kad vienetų skaičius yra nelyginis tik antrojoje eilutėje (pažymėta rodykle). Nusprendžiame, kad klaida įvyko antrosios eilutės kontroliniame simbolyje (apibrauktas), t. y. šeštojoje pozicijoje:

$$\begin{array}{cc|c|l} 1 & 1 & 0 & \\ 0 & 1 & \boxed{0} & \leftarrow \\ \hline 1 & 0 & & \end{array}$$

Ištaisę klaidą šeštojoje pozicijoje, gauname vektorių $c' = 11001110$. Matome, kad jis sutampa su $c$, t. y. klaidą ištaisėme. □

#### Klaidų taisymas

Matome, kad vieną klaidą lentelės kodas visada ištaiso.

**1.2.15 užduotis.** 1. Rasti pavyzdį, kuriame lentelės kodas dekoduotų klaidingai, kai padarytos dvi klaidos.

2. Įrodyti, kad jei padarytos dvi klaidos, tai dekoduojant tokiu būdu arba dekoduojama neteisingai, arba dekodavimas nėra vienareikšmiškas.

Taigi, šis kodas yra vieną klaidą taisantis kodas.

#### Klaidų aptikimas

**1.2.16 užduotis.** Nustatyti, kiek klaidų lentelės kodas aptinka.

#### Kodo koeficientas

Lentelės kodo koeficientas yra $\frac{4}{8} = \frac{1}{2}$.

#### Apibendrinimas

Kai kuriais atžvilgiais lentelės kodas geresnis ir už pakartojimo kodą (didesnis kodo koeficientas), ir už kontrolinio simbolio kodą (ištaiso vieną klaidą). Kitame poskyryje matysime, kad galime rasti dar geresnį vieną klaidą taisantį kodą ilgio 4 vektoriams koduoti.

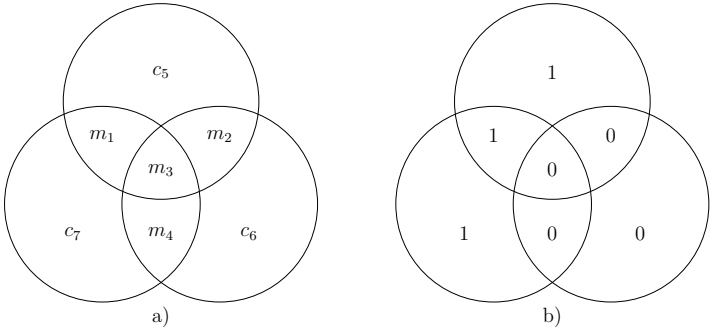
**1.2.17 užduotis.** Sudarykite lentelės kodo bendresnį variantą. Tarkime, pradinis pranešimas yra sudarytas ne iš 4, o iš $mn$ ženklų. Surašome jį į $m \times n$ lentelę. Paaiškinkite, kaip galima būtų koduoti tokiu atveju. Ar skirtųsi dekodavimo algoritmas? Kiek klaidų toks kodas galėtų ištaisyti? Koks būtų kodo koeficientas?

### 1.2.6 Hemingo kodas

Yra visa šeima dvejetainių Hemingo[2] kodų. Šiame poskyryje mes panagrinėsime tik vieną iš jų, koduojantį ilgio 4 žodžius.

#### Kodavimas

Pranešimą $m = m_1m_2m_3m_4$ koduojame žodžiu $c = m_1m_2m_3m_4c_5c_6c_7$. Kodavimą galima pavaizduoti grafiškai taip. Surašome šiuos septynis ženklus į susikertančius skritulius taip, kaip parodyta 1.1a) paveiksle. Kontroliniai simboliai $c_5, c_6, c_7$ parenkami taip, kad kiekviename skritulyje esančių vienetų skaičius būtų lyginis.




1.1 pav.: Kodavimas Hemingo kodu

**1.2.18 pavyzdys.** Jei $m = 1000$, tai $c = 1000101$, žr. 1.1b) pav. □

#### Dekodavimas

Parodysime, kad šis kodas visada ištaiso pavienę klaidą.

Tarkime, klaida įvyko pozicijoje, kuri priklauso tik vienam skrituliui, pavyzdžiui, pozicijoje $c_5$. Dekoduodami patikriname, kuriuose skrituliuose vienetų skaičius yra nelyginis. Matome, kad viršutiniame skritulyje jis nelyginis, kituose — lyginis. Padarome išvadą, kad klaida įvyko pozicijoje, kuri priklauso viršutiniam skrituliui ir nepriklauso kitiems skrituliams. Tokia pozicija tėra tik viena, ir tai būtent $c_5$, galime ištaisyti joje padarytą klaidą.

Lygiai taip pat vienareikšmiškai nustatome klaidos poziciją, jei klaida padaryta simbolyje, priklausančiame lygiai dviem iš trijų skritulių, pavyzdžiui, $m_2$ (randame, kad vienetų skaičius nelyginis viršutiniame ir dešiniajame skrituliuose, ir nusprendžiame, kad klaida padaryta pozicijoje, kuri priklauso šiems dviem skrituliams, ir nepriklauso trečiajam), ir jei klaida padaryta simbolyje, priklausančiame visiems trims skrituliams ($m_3$).

[2]Richard Hamming (1915–1998) — amerikiečių matematikas, vienas iš kodavimo teorijos pradininkų.



#### Klaidų taisymas

Taigi, kad ir kur būtų padaryta pavienė klaida, mes galime rasti jos poziciją ir ją ištaisyti.

Bet dviejų klaidų kodas jau nebeištaiso. Pavyzdžiui, tarkime, klaidos įvyko pozicijose $m_4$ ir $c_7$. Tada vienetų skaičius nelyginis pasidarys tik dešiniajame skritulyje, dėl to ištaisysime $c_6$: dekoduodami ne tik neištaisysime klaidų, bet dar daugiau jų įvelsime. Taigi, šis kodas yra vieną klaidą taisantis kodas.

**1.2.19 užduotis.** Įrodykite, kad įvykus dviems klaidoms, Hemingo kodas visada dekoduos klaidingai.

#### Klaidų aptikimas

Hemingo kodas yra 2 klaidas aptinkantis kodas, nes padarius dvi klaidas, būtinai kuriame nors skritulyje vienetų skaičius pasidarys nelyginis (patikrinkite patys, žr. 1.2.20 užduotį), iš ko ir nuspręsime, kad įvyko klaida, o įvykus trims klaidoms, vienetų skaičius visuose skrituliuose gali išlikti lyginis, ir klaidų galime nepastebėti.

**1.2.20 užduotis.**

1. Įrodykite, kad įvykus dviems klaidoms, bent viename skritulyje vienetų skaičius pasidarys nelyginis.
2. Raskite pavyzdį, kai įvykus trims klaidoms, vienetų skaičius visuose skrituliuose lieka lyginis.

#### Kodo koeficientas

Šio kodo koeficientas yra $\frac{4}{7}$.

#### Apibendrinimas

Hemingo kodas taiso irgi vieną klaidą, kaip ir lentelės kodas, bet jo kodo koeficientas yra didesnis.

Su visa Hemingo kodų šeima susipažinsime 3.1 poskyryje.

## 1.3 Kanalai ir Šenono teorema

Šiame poskyryje susipažinsime su paprasčiausiais naudojamais kanalų modeliais bei aptarsime Šenono teoremą — svarbų informacijos teorijos rezultatą, parodantį klaidas taisančių kodų teorines galimybes.

### 1.3.1 Diskretusis be atminties kanalas

Šiame poskyryje panagrinėsime bendriausią diskretaus kanalo, kuriame klaidos daromos atsitiktinai, nepriklausomai viena nuo kitos, modelį.

**1.3.1 apibrėžimas.** Diskretusis be atminties kanalas — *tai trejetas* $(A, B, \Pi)$*, kur*

- $A$ *yra aibė, sudaryta iš* $s \geqslant 1$ *elementų, vadinama* įėjimo abėcėle, $A = \{a_1, \ldots, a_s\}$,
- $B$ *yra aibė, sudaryta iš* $t \geqslant 1$ *elementų, vadinama* išėjimo abecėle, $B = \{b_1, \ldots, b_t\}$,

- $\Pi$ *yra* $s \times t$ *matrica*

$$\Pi = \begin{pmatrix} p_{11} & \dots & p_{1t} \\ \vdots & \ddots & \vdots \\ p_{s1} & \dots & p_{st} \end{pmatrix},$$

*tenkinanti savybes:*

*1) visi jos elementai yra neneigiami realūs skaičiai, t. y.* $p_{ij} \geqslant 0\ \forall i = 1, \dots, s, \forall j = 1, \dots, t$,

*2) kiekvienos eilutės elementų suma yra lygi 1, t. y.* $\sum_{j=1}^{t} p_{ij} = 1\ \forall i = 1, \dots, s$.

*Šias savybes tenkinanti matrica* $\Pi$ *vadinama* tikimybine matrica.

**1.3.2 pastaba.** Diskrečiojo be atminties kanalo apibrėžimas interpretuojamas taip. Laikome, kad kanalu siunčiami simboliai iš įėjimo abėcėlės $A$, o iš kanalo išėję simboliai priklauso išėjimo abėcėlei $B$. Dažniausiai laikysime, kad $A = B$.

Matrica $\Pi$ interpretuojama taip. Siuntėją sutapatiname su atsitiktiniu dydžiu $X$, įgyjančiu reikšmes iš abėcėlės $A$, o gavėją su atsitiktiniu dydžiu $Y$, įgyjančiu reikšmes iš abėcėlės $B$. Tada matricos elementą $p_{ij}$ interpretuojame kaip tikimybę, kad jei į kanalą pasiųstas simbolis $a_i$ (t. y. jei $X$ įgyja reikšmę $a_i$), tai iš kanalo išėjo simbolis $b_j$ (t. y. $Y$ įgijo reikšmę $b_j$), t. y. $p_{ij}$ — tai sąlyginė tikimybė $p_{ij} = P(Y = b_j | X = a_i)$. Trumpiau ją žymėsime $P(b_j|a_i)$ — tikimybė, kad iš kanalo išeis $b_j$, jei į kanalą įėjo $a_i$.

**1.3.3 pavyzdys.** Diskretusis be atminties kanalas $(A, B, \Pi)$, kur $A = \{0, 1\}$, $B = \{0, 1\}$,

$$\Pi = \begin{pmatrix} 0{,}9 & 0{,}1 \\ 0{,}2 & 0{,}8 \end{pmatrix}.$$

Jei į šį kanalą pasiųsime 0, tai su tikimybe 0,9 iš kanalo išeis 0, o su tikimybe 0,1 iš kanalo išeis 1. Jei pasiųsime 1, tai su tikimybe 0,8 iš kanalo išeis 1, o su tikimybe 0,2 išeis 0. □

Apibrėžime „be atminties“ reiškia, kad simboliai iškraipomi nepriklausomai vienas nuo kito. Taip yra todėl, kad $\Pi$ yra nekintantis dydis (iškraipymo tikimybės nekinta laikui bėgant). „Diskretusis“ reiškia, kad įėjimo ir išėjimo aibės yra diskrečiosios.

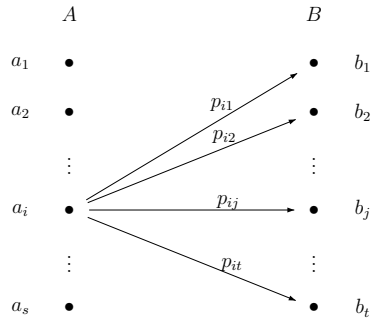
Diskretųjį be atminties kanalą galima pavaizduoti grafiškai taip, kaip parodyta 1.2 pav. Kairėje pusėje parodyti aibės $A$ elementai, dešinėje — aibės $B$ elementai. Rodyklės su virš jų pažymėtais matricos $\Pi$ elementais parodo, su kokia tikimybe iš kanalo gali išeiti aibės $B$ elementas, jei į kanalą pasiunčiamas aibės $A$ elementas. Pavyzdžiui, rodyklė iš $a_i$ į $b_j$ rodo, kad jei į kanalą pasiunčiamas aibės $A$ elementas $a_i$, tai su tikimybe $p_{ij}$ iš kanalo gali išeiti $b_j$. Aišku, tokios rodyklės turi eiti iš kiekvieno aibės $A$ elemento į kiekvieną aibės $B$ elementą. Tačiau šioje diagramoje, kad jos neperkrautume detalėmis, palikome tik rodykles, einančias iš $a_i$.

Tolesniuose poskyriuose panagrinėsime kelis naudingus diskrečiojo be atminties kanalo pavyzdžius.

### 1.3.2 Dvejetainis simetrinis kanalas

Tai vienas iš paprasčiausių diskrečiojo be atminties kanalo pavyzdžių.






1.2 pav.: Diskretusis be atminties kanalas

**1.3.4 apibrėžimas.** *Tegu* $0 \leqslant p \leqslant 1$. Dvejetainis simetrinis kanalas su iškraipymo tikimybe $p$ — *tai diskretusis be atminties kanalas* $(A, B, \Pi)$, *kuriame* $A = B = \{0, 1\}$ *ir*

$$\Pi = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}.$$

Iš apibrėžimo matome, kad dvejetainio simetrinio kanalo tikimybės yra tokios:

$$P(1|0) = P(0|1) = p,$$
$$P(0|0) = P(1|1) = 1 - p.$$

Apibrėžime „dvejetainis“ reiškia, kad kanalo abėcėlės yra dvinarės. „Simetrinis“ reiškia, kad iškraipymo tikimybės nepriklauso nuo to, kuris ženklas įeina į kanalą — ar tai būtų 0, ar 1, tikimybė, kad jis pats išeis iš kanalo, yra $1-p$, o tikimybė, kad ne jis išeis, yra $p$. Taigi, dvejetainio simetrinio kanalo tikimybinė matrica yra simetrinė pagrindinės įstrižainės atžvilgiu.

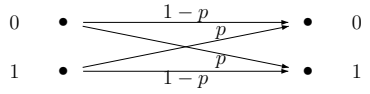
**1.3.5 pavyzdys.** Jei į dvejetainį simetrinį kanalą su iškraipymo tikimybe 0,1 pasiųsime 0, tai su tikimybe 0,9 iš kanalo išeis 0, o su tikimybe 0,1 iš kanalo išeis 1. Jei pasiųsime 1, tai su tikimybe 0,9 iš kanalo išeis 1, o su tikimybe 0,1 išeis 0. Šio kanalo tikimybinė matrica yra

$$\Pi = \begin{pmatrix} 0{,}9 & 0{,}1 \\ 0{,}1 & 0{,}9 \end{pmatrix}.$$ □

Dvejetainį simetrinį kanalą su iškraipymo tikimybe $p$ grafiškai galima pavaizduoti taip, kaip parodyta 1.3 pav.

### 1.3.3 $q$-tainis simetrinis kanalas

Tai dvejetainio simetrinio kanalo apibendrinimas abėcėlei iš $q$ elementų.




1.3 pav.: Dvejetainis simetrinis kanalas su iškraipymo tikimybe $p$

**1.3.6 apibrėžimas.** *Tegu* $0 \leqslant p \leqslant 1$, $q \geqslant 2$. $q$-tainis simetrinis kanalas su iškraipymo tikimybe $p$ *— tai diskretusis be atminties kanalas* $(A, B, \Pi)$*, kuriame* $A = B$, $|A| = q$ *ir*

$$\Pi = \begin{pmatrix} 1-p & \frac{p}{q-1} & \frac{p}{q-1} & \cdots & \frac{p}{q-1} \\ \frac{p}{q-1} & 1-p & \frac{p}{q-1} & \cdots & \frac{p}{q-1} \\ \frac{p}{q-1} & \frac{p}{q-1} & 1-p & \cdots & \frac{p}{q-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{p}{q-1} & \frac{p}{q-1} & \frac{p}{q-1} & \cdots & 1-p \end{pmatrix}.$$

Taigi, $q$-tainio simetrinio kanalo tikimybės yra tokios:

$$P(b|a) = \begin{cases} 1-p & \text{, jei } a = b; \\ \frac{p}{q-1} & \text{, jei } a \neq b. \end{cases}$$

**1.3.7 pastaba.** Iš kur atsiranda $\frac{p}{q-1}$? Iškraipymo tikimybė $p$ yra tikimybė, kad jei į kanalą pasiųsime $a \in A$, tai iš kanalo išeis ne $a$, o bet kuris iš $q-1$ likusių išėjimo abėcėlės $B$ elementų. Tą iškraipymo tikimybę po lygiai padalijame kiekvienam aibės $B$ elementui, nelygiam $a$, todėl tikimybė, kad iš kanalo išeis kažkoks konkretus $b \in B$, yra $\frac{p}{q-1}$.

**1.3.8 pavyzdys.** Trejetainis simetrinis kanalas su iškraipymo tikimybe 0,1, kurio įėjimo ir išėjimo abėcėlės yra $A = B = \{0, 1, 2\}$. Jo tikimybinė matrica yra

$$\Pi = \begin{pmatrix} 0{,}9 & 0{,}05 & 0{,}05 \\ 0{,}05 & 0{,}9 & 0{,}05 \\ 0{,}05 & 0{,}05 & 0{,}9 \end{pmatrix}.$$

Jei į šį kanalą pasiųsime 0, tai su tikimybe 0,9 iš kanalo išeis 0, su tikimybe 0,05 iš kanalo išeis 1 ir su tikimybe 0,05 išeis 2. Analogiškai jei pasiųstume 1 ir 2. □

Jei $A = \{0, 1, \ldots, q-1\}$, tai $q$-tainį simetrinį kanalą su iškraipymo tikimybe $p$ grafiškai galima pavaizduoti taip, kaip parodyta 1.4 pav.

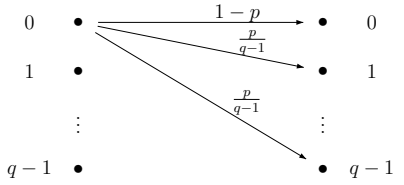
### 1.3.4 Dvejetainis simetrinis trinantis kanalas

Kartais sunku nuspręsti, koks dvinarės abėcėlės simbolis išėjo iš kanalo. Tokiam atvejui modeliuoti tinka dvejetainis simetrinis trinantis kanalas.

**1.3.9 apibrėžimas.** *Tegu* $0 \leqslant p \leqslant 1$, $0 \leqslant r \leqslant 1$. Dvejetainis simetrinis trinantis kanalas su iškraipymo tikimybe $p$ ir ištrynimo tikimybe $r$ *— tai diskretusis be atminties kanalas, kuriame* $A = \{0, 1\}$, $B = \{0, 1, ?\}$ *ir*

$$\Pi = \begin{pmatrix} 1-p-r & p & r \\ p & 1-p-r & r \end{pmatrix}.$$






1.4 pav.: $q$-tainis simetrinis kanalas su iškraipymo tikimybe $p$

Čia klaustukas žymi tai, kad pasiųstas ženklas išsitrynė. Yra tik žinoma, kad toje vietoje buvo kažkoks ženklas, o kuris — ar nulis, ar vienetas, — neaišku.

Dvejetainio simetrinio trinančio kanalo tikimybės yra tokios:

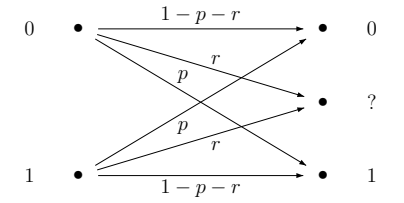
$$P(0|0) = P(1|1) = 1 - p - r,$$
$$P(1|0) = P(0|1) = p,$$
$$P(?|0) = P(?|1) = r.$$

**1.3.10 pavyzdys.** Dvejetainio simetrinio kanalo su iškraipymo tikimybe 0,1 ir ištrynimo tikimybe 0,2 tikimybinė matrica yra

$$\Pi = \begin{pmatrix} 0{,}7 & 0{,}1 & 0{,}2 \\ 0{,}1 & 0{,}7 & 0{,}2 \end{pmatrix}.$$

Jei į šį kanalą pasiųsime 0, tai su tikimybe 0,7 iš kanalo išeis 0, su tikimybe 0,1 iš kanalo išeis 1 ir su tikimybe 0,2 išeis „?“. Analogiškai, jei pasiųstume 1. □

Dvejetainį simetrinį trinantį kanalą su iškraipymo tikimybe $p$ ir ištrynimo tikimybe $r$ grafiškai galima pavaizduoti taip, kaip parodyta 1.5 pav.




1.5 pav.: Dvejetainis simetrinis trinantis kanalas su iškraipymo tikimybe $p$ ir ištrynimo tikimybe $r$

### 1.3.5 Grupė

Gali atrodyti, kad sudėtingų matematinių sąvokų naudojimas tik komplikuoja situaciją. Gal kartais taip ir būna. Bet dažniausiai tai leidžia, panaudojus jau žinomus matematinius rezultatus,

gauti naujus rezultatus tiriamoje srityje. Mes šiame kurse irgi panaudosime nemažai matematinių (daugiausiai algebros) sąvokų, leisiančių sukonstruoti gerus kodus. Tas sąvokas prisiminsime keliose vietose. Šiame poskyryje prisiminsime operacijos, grupės, Abelio grupės apibrėžimus. Tai mums leis aprūpinti kanalo abėcėles Abelio grupės struktūra, kas savo ruožtu leis kanalo poveikį užrašyti į kanalą pasiųsto vektoriaus ir klaidų vektoriaus suma (žr. kitą poskyrį). Paskui 2.1 poskyryje prisiminsime kūnų ir tiesinių erdvių sąvokas, o 4.1 poskyryje — polinomų žiedų sąvokas.

**1.3.11 apibrėžimas.** *Tarkime, aibė $A$ yra netuščia.* Algebrine operacija *(arba tiesiog* operacija*) aibėje $A$ vadinama dviejų argumentų funkcija, apibrėžta aibėje $A$ ir įgyjanti reikšmes iš tos pačios aibės, t. y. funkcija iš $A \times A$ į $A$.*

**1.3.12 pavyzdžiai.** 1. Aibėje $\mathbb{F}_2 = \{0, 1\}$ galime apibrėžti sudėties operaciją $+$ tokia lentele:

| $+$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Iš lentelės matome, kad, pavyzdžiui, $1 + 1 = 0$. Tokiu būdu apibrėžta sudėties operacija sutampa su sveikųjų skaičių sudėties moduliu 2 operacija aibėje $\{0, 1\}$.

2. Aibėje $\mathbb{F}_2 = \{0, 1\}$ galime apibrėžti daugybos operaciją $\cdot$ tokia lentele:

| $\cdot$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Iš lentelės matome, kad, pavyzdžiui, $1 \cdot 1 = 1$. Tokiu būdu apibrėžta daugybos operacija sutampa su sveikųjų skaičių daugybos moduliu 2 operacija aibėje $\{0, 1\}$.

3. Tegu $p$ — pirminis skaičius. Aibėje $\mathbb{F}_p = \{0, 1, \dots, p-1\}$ galime apibrėžti sudėties ir daugybos moduliu $p$ operacijas.

4. Tegu $p$ — pirminis skaičius. Pažymėkime $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\} = \{1, \dots, p-1\}$. Ar galime aibėje $\mathbb{F}_p^*$ apibrėžti sudėties moduliu $p$ operaciją? O daugybos moduliu $p$ operaciją? Kad būtų galima apibrėžti, rezultatas turi irgi priklausyti aibei $\mathbb{F}_p^*$.

   a) Ar $(a + b) \bmod p \in \mathbb{F}_p^*$ visiems aibės $\mathbb{F}_p^*$ elementams $a$ ir $b$? Ne, pavyzdžiui, jei $a = 1$, $b = p - 1$, tai
   $$(a + b) \bmod p = (1 + p - 1) \bmod p = p \bmod p = 0,$$
   o 0 nepriklauso aibei $\mathbb{F}_p^*$. Taigi, sudėties moduliu $p$ operacijos aibėje $\mathbb{F}_p^*$ apibrėžti negalime.

   b) Ar $(a \cdot b) \bmod p \in \mathbb{F}_p^*$ visiems aibės $\mathbb{F}_p^*$ elementams $a$ ir $b$? t. y. ar galime gauti nulį, sudauginę aibės $\mathbb{F}_p^*$ elementus $a$ ir $b$? Nesunku pamatyti, kad negalime. Iš tikrųjų, nulį moduliu $p$ gausime tik tokiu atveju, jei sandauga $ab$ (čia $a$ ir $b$ sudauginame, kaip sveikuosius skaičius) yra $p$ kartotinis, $ab = ps$, čia $s$ yra sveikasis skaičius, $s \geqslant 1$. Paskutinė lygybė negali būti teisinga, nes dešinioji jos pusė dalinasi iš pirminio skaičiaus $p$, o kairioji dalintis negali, nes $a$ ir $b$ yra už $p$ mažesni skaičiai. Taigi, galime apibrėžti daugybos moduliu $p$ operaciją aibėje $\mathbb{F}_p^*$. □



**1.3.13 apibrėžimas.** *Tarkime, aibė $A$ yra netuščia, ir joje yra apibrėžta* operacija $\diamond$. *Rinkinys $(A, \diamond)$ (t. y. aibė $A$ kartu su operacija $\diamond$) vadinamas* grupe*, jei:*

1. *Operacija $\diamond$ yra asociatyvi, t. y.*
   $$\forall a, b, c \in A \quad (a \diamond b) \diamond c = a \diamond (b \diamond c),$$
2. *Operacija $\diamond$ turi neutralųjį elementą, t. y. aibėje $A$ yra toks elementas $i$, kad*
   $$\forall a \in A \quad i \diamond a = a \diamond i = a,$$
3. *Kiekvienas aibės $A$ elementas $a$ turi toje pačioje aibėje elementą $\overline{a}$, simetrišką operacijos $\diamond$ atžvilgiu, t. y. tokį, kad*
   $$a \diamond \overline{a} = \overline{a} \diamond a = i.$$

*Jeigu, be to,*

4. *Operacija $\diamond$ yra komutatyvi, t. y.*
   $$\forall a, b \in A \quad a \diamond b = b \diamond a,$$

*tai grupė vadinama* komutatyviąja*, arba* Abelio*, grupe.*

**1.3.14 pastabos.** 1. Šiame apibrėžime $\diamond$ žymi bet kokią operaciją 1.3.11 apibrėžimo prasme. Ji gali būti vadinama sudėtimi, daugyba ir t. t.

2. Dažnai sakoma „aibė $A$ yra grupė operacijos $\diamond$ atžvilgiu“.

3. Jei aišku, apie kurią operaciją eina kalba, vietoj „grupė $(A, \diamond)$“ sakysime tiesiog „grupė $A$“.

**1.3.15 pastabos.** 1. Jeigu algebrinė operacija grupėje $A$ vadinama *sudėtimi* ir žymima $+$, tai jos neutralusis elementas vadinamas *nuliniu elementu* arba *nuliu* ir žymimas 0, o elementui $a$ simetriškas elementas $\overline{a}$ vadinamas jam *priešingu* elementu ir žymimas $-a$ (t. y. $\overline{a} = -a$). Be to, galima apibrėžti naują operaciją, vadinamą *atimtimi* ir žymimą $-$, tokiu būdu: elementų $a$ ir $b$ *skirtumas* $a - b$ yra $a + (-b)$, t. y. $a - b = a + (-b)$. Grupė sudėties atžvilgiu vadinama *adicine* grupe.

2. Analogiškai, kai operacija vadinama *daugyba* ir žymima $\times$ arba tašku $\cdot$, kuris gali būti ir praleidžiamas, tai jos neutralusis elementas vadinamas *vienetiniu elementu* arba *vienetu* ir žymimas 1, o elementui $a$ simetriškas elementas $\overline{a}$ vadinamas jam *atvirkštiniu* elementu ir žymimas $a^{-1}$ arba $\frac{1}{a}$ (t. y. $\overline{a} = a^{-1}$). Be to, galima apibrėžti naują operaciją, vadinamą *dalyba* ir žymimą / arba :, tokiu būdu: elementų $a$ ir $b$ *santykis* $a/b$ yra $a \cdot b^{-1}$, t. y. $a/b = a \cdot b^{-1}$. Grupė daugybos atžvilgiu vadinama *multiplikacine* grupe.

**1.3.16 pavyzdžiai.** Tarkime, $p$ yra pirminis skaičius.

1. Nesunkiai galite patikrinti, kad $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ (tuo pačiu ir $\mathbb{F}_2 = \{0, 1\}$) yra Abelio grupė sudėties moduliu $p$ operacijos, apibrėžtos 1.3.12 pavyzdžio 3 dalyje, atžvilgiu. Jos neutralus elementas yra 0. Nuliui priešingas elementas yra jis pats (t. y. $-0 = 0$), o bet kurio nelygaus nuliui elemento $a$ priešingas elementas $-a$ yra $p - a$. Pavyzdžiui, grupėje $\mathbb{F}_3 = \{0, 1, 2\}$ turime, kad $-1 = 3 - 1 = 2$ ir $-2 = 3 - 2 = 1$. Skirtumas $1 - 2 = 1 + (-2) = 1 + 1 = 2$ (kadangi skaičiuojame moduliu 3, tą patį skirtumą galime skaičiuoti ir kitaip: $1 - 2 = -1 = 2$). Kitas pavyzdys: grupėje $\mathbb{F}_2 = \{0, 1\}$ turime, kad $-1 = 2 - 1 = 1$.

2. $\mathbb{F}_p^* = \{1, \ldots, p-1\}$ (tuo pačiu ir $\mathbb{F}_2^* = \{1\}$) yra Abelio grupė daugybos moduliu $p$ operacijos, apibrėžtos 1.3.12 pavyzdžio 4 dalyje, atžvilgiu. Jos neutralus elementas yra 1. Vienetui atvirkštinis elementas yra jis pats (t. y. $1^{-1} = 1$), o kitiems elementams atvirkštinius rasti nėra taip paprasta. Mes pasitenkinsime paprasčiausiu perrinkimu. Pavyzdžiui, grupėje $\mathbb{F}_5^* = \{1, 2, 3, 4\}$ atvirkštinis dvejetui yra 3 (t. y. $2^{-1} = 3$), nes šioje grupėje $2 \cdot 3 = 1$ (iš tikro, $(2 \cdot 3) \bmod 5 = 6 \bmod 5 = 1$), atvirkštinis trejetui yra 2 (t. y. $3^{-1} = 2$), o atvirkštinis ketvertui yra 4 (t. y. $4^{-1} = 4$). Santykis $4/3$ bus lygus 3, nes $4/3 = 4 \cdot 3^{-1} = 4 \cdot 2 = 3$ (nes skaičiuojame moduliu 5).

3. $\mathbb{F}_p = \{0, 1, \ldots, p-1\}$ nėra grupė daugybos moduliu $p$ operacijos atžvilgiu, nes nulis neturi atvirkštinio elemento (t. y. nėra tokio elemento, iš kurio padauginę nulį gautume vienetą). □

### 1.3.6 Adityvusis kanalas

Kartais naudinga mokėti elementus sudėti, atimti, o ir neutralus elementas — nulis — būna ne pro šalį. Tam mes aprūpinkime kanalo abėcėles adicinės Abelio grupės struktūra. Tada galėsime apibrėžti klaidų vektorių kaip skirtumą tarp iš kanalo išėjusio vektoriaus ir į kanalą pasiųsto žodžio.

**1.3.17 apibrėžimas.** *Diskretųjį be atminties kanalą* $(A, B, \Pi)$ *vadinsime* adityviuoju*, jei* $A = B$ *ir* $A$ *— baigtinė Abelio grupė sudėties atžvilgiu.*

Adityviojo kanalo abėcėlių elementus galima sudėti, atimti. Nuo šiol laikysime, kad mūsų naudojamas kanalas yra adityvusis.

Kaip įprasta, $A^n$ žymėsime aibę visų $n$ ilgio vektorių, kurių koordinatės priklauso aibei $A$.

**1.3.18 apibrėžimas.** *Tarkime, į adityvųjį kanalą įeina vektorius* $c \in A^n$*, o išeina vektorius* $y \in A^n$*. Tada vektorių* $e = y - c$ *vadinsime* klaidų vektoriumi. Klaidų padėtimis *vadinsime klaidų vektoriaus nenulinių koordinačių pozicijas.* Klaidų reikšmės *yra klaidų vektoriaus koordinatės.*

**1.3.19 pavyzdys.** Tarkime, adityviojo trejetainio kanalo abėcėlė yra $A = \mathbb{F}_3 = \{0, 1, 2\}$, kur veiksmai atliekami moduliu 3. Tarkime, kad į kanalą įeina vektorius $c = 12010$, o išeina vektorius $y = 10210$. Tada klaidų vektorius yra $e = y - c = 01200$, klaidų padėtys yra 2 ir 3, o klaidų reikšmės yra tokios: pirmoje pozicijoje įvykusios klaidos reikšmė yra 0 (klaidos nebuvo), antroje — 1, trečioje — 2 ir t. t. Paprastai nulinės klaidų reikšmės nevardijamos, todėl užtenka pasakyti, kad antroje pozicijoje įvykusios klaidos reikšmė yra 1, o trečioje — 2. □

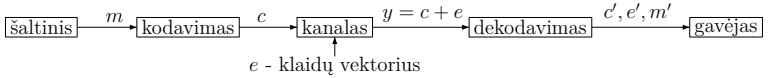
Tarkime, į adityvųjį kanalą įeina vektorius $c \in A^n$, o išeina vektorius $y \in A^n$. Pagal klaidų vektoriaus $e$ apibrėžimą, $e = y - c$, todėl $y = c + e$. Taigi, adityviojo kanalo atliekamus iškraipymus matematiškai galima užrašyti taip:

> prie į kanalą pasiųsto vektoriaus $c$ pridedamas klaidų vektorius $e \in A^n$, ir iš kanalo išeina rezultatas $y = c + e$.



### 1.3.7 Šenono teorema

Prisiminkime, kad kodavimas vyksta pagal tokią schemą:




Norimą perduoti informaciją verčiame $k$ ilgio vektoriais iš abėcėlės $A$ simbolių. Kiekvieną tokį vektorių $m$ užkoduojame, gauname didesnės apimties $n$, kur $n \geqslant k$, ilgio užkoduotą vektorių $c$. Jį siunčiame kanalu, kur galimi iškraipymai. Iš kanalo išeina $n$ ilgio vektorius $y = c + e$. Dekoduojant paprastai randami 3 dydžiai: klaidų vektorius $e'$, pataisyti užkoduotas vektorius $c'$ ir pradinis pranešimas $m'$.

Taigi, norėdami perduoti $k$ informacijos simbolių, iš tikro perduodame $n$ simbolių. Santykį $k/n$ vadiname *kodo koeficientu*. Kuo daugiau padidėja pranešimo apimtis koduojant, t. y. kuo kodo koeficientas mažesnis, tuo labiau galime sumažinti klaidingo dekodavimo tikimybę, bet tuo pačiu tuo labiau apkrauname kanalą (reikia persiųsti daugiau simbolių). Norėtume sumažinti klaidingo dekodavimo tikimybę, tuo pačiu išlaikydami kodo koeficientą pakankamai didelį. 1948 m. Klodas Šenonas[3] įrodė savo žymiąją teoremą, kurios esmė yra tokia:

> mes galime sumažinti klaidingo dekodavimo tikimybę tiek, kiek tik norime, naudodami pakankamai ilgus kodus, jei tik jų kodo koeficientas neviršija tam tikro dydžio, vadinamo diskrečiojo be atminties kanalo *talpa*.

Šenono teorema rodo, kad klaidingo dekodavimo tikimybę galime padaryti kiek norime mažą, tuo pačiu išlaikydami kodo koeficientą gana didelį — artimą kanalo talpai.

Deja, šios teoremos įrodymas nėra konstruktyvus, jis neparodo, kaip sudaryti tuos „gerus“ kodus. Be to, „geras“ kodas nebūtinai bus praktiškas, nes galbūt kodavimas ir dekodavimas pareikalaus pernelyg daug atminties ir laiko resursų. „Gerus“ praktiškus kodus gauti ir yra klaidas taisančių kodų teorijos tikslas.

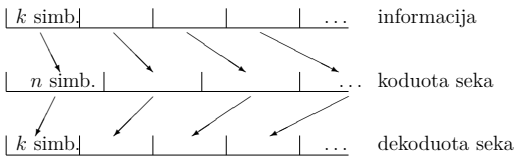
## 1.4 Sąsūkų kodai

Klaidas taisančius kodus galima suskirstyti į dvi grupes: blokinius ir sąsūkų (konvoliucinius, angl. *convolutional*). Iki šiol pavyzdžiuose mes nagrinėjome tik blokinius kodus. Šiame poskyryje palyginsime šias dvi kodų grupes, trumpai (pavyzdžiu) pristatysime sąsūkų kodus, ir šiame kurse jų daugiau nebeliesime.

[3] Claude Shannon (1916–2001) — amerikiečių matematikas, laikomas informacijos teorijos pradininku. Jo 1948 metų straipsnis „A Mathematical Theory of Communication“, išspausdintas „The Bell System Technical Journal“, laikomas pirmuoju ne tik informacijos teorijos, bet ir klaidas taisančių kodų teorijos straipsniu. Be to, K. Šenonas žymus ir kitais savo darbais. Jis laikomas ir skaitmeninės skaičiavimo mašinos (kompiuterio) teorijos kūrėju, nes 1937 metais savo magistro darbe įrodė, kad elektros grandinės gali realizuoti bet kokius logikos ir skaičių sąryšius. Taip pat jis įnešė svarbų indėlį ir į kriptografijos teoriją savo Antrojo pasaulinio karo metu atliktais darbais.

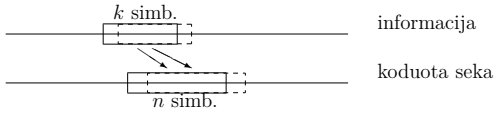
### 1.4.1 Blokinių ir sąsūkų kodų palyginimas

Kai naudojame blokinius kodus, tai informaciją skaidome į $k$ ilgio blokus ir kiekvieną jų, nepriklausomai vieną nuo kito, užkoduojame $n$ ilgio blokais, o paskui taip pat nepriklausomai kiekvieną ilgio $n$ bloką dekoduojame (žr. 1.6 pav.).




1.6 pav.: Blokiniai kodai

Tuo tarpu sąsūkų kodai veikia kitaip. Turime informacijos srautą, kuriame bet kuriuo laiko momentu užkoduota informacija priklauso nuo $k$ paskutinių informacijos simbolių (žr. 1.7 pav.). Kodavimas ir dekodavimas vyksta tolydžiai.




1.7 pav.: Sąsūkų kodai

Sąsūkų kodų privalumai, lyginant su blokiniais:

1. Greitas kodavimas ir dekodavimas, vykstantys vienu metu su informacijos siuntimu ir gavimu;
2. Nekyla problemų dėl sinchronizacijos, išskyrus pačią perdavimo pradžią (tuo tarpu koduojant blokiniais kodais reikia nustatyti kiekvieno bloko pradžią ir pabaigą);
3. Kodavimo ir dekodavimo schemos labai paprastos, tuo tarpu blokiniams kodams sunku rasti efektyvius dekodavimo algoritmus (didelei daliai blokinių kodų efektyvūs dekodavimo algoritmai išvis neegzistuoja).

Sąsūkų kodų trūkumai, lyginant su blokiniais:

1. Mažesnės klaidų ištaisymo galimybės nei blokiniame kodavime;
2. Jų struktūra mažiau ištyrinėta, tuo tarpu blokinių kodų struktūra yra įvairesnė ir geriau išstudijuota, nes blokiniai kodai yra artimesni tradicinėms, gerai ištyrinėtoms matematinėms struktūroms.



### 1.4.2 Sąsūkų kodų kodavimas

Šiame ir tolimesniuose poskyriuose parodysime pavyzdžiu, kaip veikia sąsūkos kodų kodavimas ir dekodavimas.

Tarkime turime begalinę dvejetainę matricą $G$:

$$G = \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1} & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

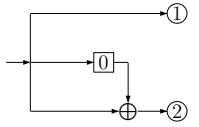
Begalinį informacijos vektorių $m$ užkoduojame daugindami iš matricos $G$. Pavyzdžiui, jei

$$m = (1\,0\,1\,1\,0\,0\ldots),$$

tai jis užkoduojamas vektoriumi

$$c = m \cdot G = (1\,1\,0\,1\,1\,1\,1\,0\,0\,1\,0\,0\ldots).$$

Tokį kodavimą paprasčiau galime realizuoti, naudodami schemas, vadinamas *stumiamaisiais registrais* (angl. *shift register*). Mūsų pavyzdžio atveju gauti $c$ galime naudodami tokį registrą:




Čia stačiakampis yra atmintis, užlaikanti įrašytą reikšmę vieną laiko momentą. Pradiniu laiko momentu jos reikšmė yra nulis.

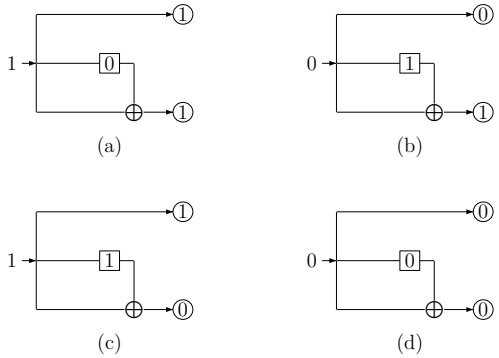
Į šią schemą rodyklės kryptimi įėjus vienam simboliui, išeina du. Pirmasis iš jų (pažymėtas skaičiumi 1) yra lygus įėjusiam (matome, kad kas tik įeina, tuoj viršutine atšaka ir išeina). Antrasis (pažymėtas skaičiumi 2) yra suma (moduliu 2) įėjusio simbolio ir iš atminties išėjusio simbolio, t. y. simbolio, įėjusio prieš tai. Taigi, antrasis yra paskutinio ir priešpaskutinio įėjusių simbolių suma. Be to, įėjęs simbolis nukeliauja ir į atmintį (vidurinė atšaka), iš kurios išeis kitu laiko momentu (įėjus kitam simboliui).

Panagrinėkime, kaip iš $m$ gauname $c$ naudodami duotą schemą. 1.8 paveikslo brėžiniuose parodoma, kokie simboliai išeina, priklausomai nuo to, kas įėjo ir kas buvo atmintyje. Matome, kad pradžioje įeina 1, atmintyje yra 0, todėl išeina 11 (schema (a)). Tada įeina 0, atmintyje — 1, išeina 01 (schema (b)). Toliau kartojasi pirma situacija: įeina 1, atmintyje — 0, išeina 11. Tada įeina 1, atmintyje 1, išeina 10 (schema (c)). Tuomet vėl kartojasi: įeina 0, atmintyje 1, išeina 01. Tada įeina 0, atmintyje 0, išeina 00 (schema (d)). Ir t.t.

Gauname tokį užkoduotą vektorių $c$:

$$c = (\underbrace{1\,1}_{\text{(a)}}\,\underbrace{0\,1}_{\text{(b)}}\,\underbrace{1\,1}_{\text{(a)}}\,\underbrace{1\,0}_{\text{(c)}}\,\underbrace{0\,1}_{\text{(b)}}\,\underbrace{0\,0}_{\text{(d)}}\ldots)$$

Matome, kad naudodami stumiamąjį registrą, gavome lygiai tokį patį vektorių $c$, kaip ir daugindami iš matricos $G$. Kodėl taip yra?




1.8 pav.: Kodavimo sąsūkų kodu pavyzdys

Matricos $G$ nelyginiuose stulpeliuose yra vienetiniai (turintys vienetą vienoje iš pozicijų ir nulius visose kitose) vektoriai, o lyginiuose stulpeliuose yra lygiai po du vienetus (išskyrus antrą stulpelį), vadinasi jei $m = (m_1, m_2, m_3, \ldots)$, tai vektoriaus $c = (c_1, c_2, c_3 \ldots)$, gauto sudauginus $m$ su $G$, koordinatės bus tokios:

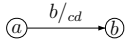
$$\begin{cases} c_{2k-1} = m_k, \\ c_{2k} = (m_{k-1} + m_k) \bmod 2, \end{cases} \quad \text{kur } k \geqslant 1,\ m_0 = 0.$$

Kaip matėme, duotas stumiamasis registras konstruoja būtent tokį vektorių $c$.

### 1.4.3 Sąsūkų kodų būsenų diagrama

Galima nubrėžti sąsūkų kodo *būsenų diagramą*. Sąsūkų kodo *būsena* vadinsime jo stumiamojo registro atminties turinį.

Tarkime, kodo būsena yra $a$, į schemą įeina koks nors simbolis $b$, išeina simboliai $c$ ir $d$, būsena keičiasi į $b$. Tai žymėsime tokiu būdu:




Tada duoto kodo būsenų diagrama parodyta 1.9 paveiksle.

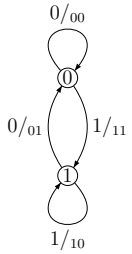
Šita diagrama leidžia lengvai koduoti. Užkoduota seka atitinka kelią šiame grafe.

### 1.4.4 Sąsūkų kodų dekodavimas

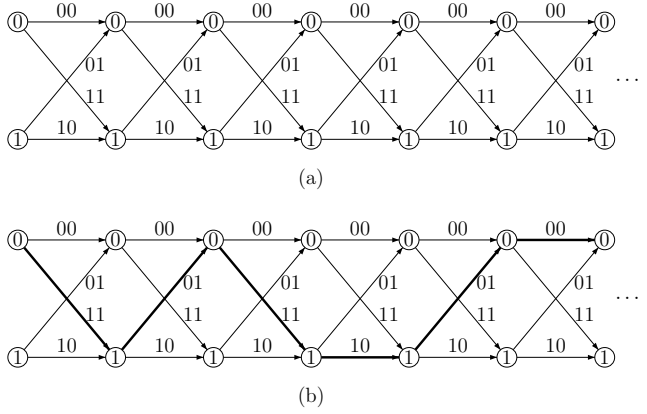
Būsenų diagramą išplečiame laike. Gauname diagramą, parodytą 1.10 paveikslo (a) dalyje.

Jei persiunčiant vektorių kanalu klaidų nepadaroma — lengvai grafe galima atsekti kelią ir dekoduoti žodį, pradedant nuo nulinės būsenos ir einant briaunomis, kurios atitinka iš kanalo






1.9 pav.: Būsenų diagrama




1.10 pav.: Sąsūkos kodų dekodavimo pavyzdys

gautą vektorių $y$. Pavyzdžiui, 1.10 paveikslo (b) dalyje pažymėtas kelias, gaunamas dekoduojant vektorių $y = (11\ 01\ 11\ 10\ 01\ 00\ldots)$. Tuo keliu einant aplankytos viršūnės ir sudarys dekoduotą vektorių $m' = (101100\ldots)$.

O kaip ištaisyti padarytas klaidas?

Iš anksto pasirenkame $t \geqslant 1$. Imame iš kanalo gauto vektoriaus $y$ dalį, sudarytą iš $t$ blokų, einančių iš eilės, lyginame ją su galimais keliais ir renkamės tą kelią, kuris mažiausiai skiriasi.

Imkime pavyzdį. Tegu užkoduotas vektorius

$$c = (\underbrace{1\,1}_{1}\ \underbrace{0\,1}_{0}\ \underbrace{1\,1}_{1}\ \underbrace{1\,0}_{1}\ \underbrace{0\,1}_{0}\ldots)$$

siunčiamas į kanalą. Tarkime, kanale iškraipyta antroji vektoriaus $c$ pozicija, ir gavėjas iš kanalo gauna tokį vektorių:

$$y = (1\ \underline{0}\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ldots).$$

Tegu $t = 2$. Imame vektoriaus $y$ pirmus $t$ blokų, t. y. vektorių $u = (1\ 0\ 0\ 1)$, ir randame atstumą (t. y. besiskiriančių koordinačių skaičių) iki kiekvieno galimo vektoriaus, gauto iš būsenų diagramos (išplėstos laike). Iš būsenų diagramos matome, kad galimi 4 vektoriai (einame iš nulinės būsenos ir žiūrime, kokie vektoriai galėjo būti gauti koduojant):

$$\begin{array}{ll} (0\ 0\ 0\ 0) & 2 \\ (0\ 0\ 1\ 1) & 2 \\ (1\ 1\ 0\ 1) & 1 \\ (1\ 1\ 1\ 0) & 3 \end{array}$$

Čia prie kiekvieno vektoriaus nurodytas jo atstumas iki $u$. Mažiausias atstumas yra trečiojo iš galimų vektorių, todėl juo keičiame $u$.

Tada dekoduojam pirmąjį bloką, t. y. 11. Dekodavę gauname 1.

Dabar kartojame viską su nauju $u$, pradedant nuo antrojo bloko. Vėl imame $t$ blokų: $u = 0111$. Būsenų diagramoje esame antro laiko momento būsenoje 1. Iš ten irgi yra keturi keliai, iš kurių renkamės artimiausią (šiuo atveju 0111) ir dekoduojame antrąjį bloką 01 į 0. Ir t.t.

Taigi, vieną klaidą ištaisė.

Dabar tarkime, kad yra dvi klaidos — antroje ir ketvirtoje vektoriaus pozicijose, t. y.

$$y = (1\ \underline{0}\ 0\ \underline{0}\ 1\ 1\ 1\ 0\ 0\ 1\ldots)$$

Imame pirmąjį vektorių $u = (1\ 0\ 0\ 0)$ ir randame atstumą iki kiekvieno galimo vektoriaus. Pagal būsenų diagramą galimi vektoriai bei atstumai nuo $u$ iki kiekvieno iš jų yra:

$$\begin{array}{ll} (0\ 0\ 0\ 0) & 1 \\ (0\ 0\ 1\ 1) & 3 \\ (1\ 1\ 0\ 1) & 2 \\ (1\ 1\ 1\ 0) & 2 \end{array}$$

Mažiausias atstumas yra iki pirmojo iš galimų vektorių, todėl juo keičiame pirmąjį vektorių. Gautą pirmąjį bloką 00 dekoduojame 0. Toliau imame antrąjį vektorių $u = (0\ 0\ 1\ 1)$, jis yra tarp galimų vektorių, todėl antrojo vektoriaus nekeičiame, antrą bloką 00 dekoduojame 0. Taip pat nekeičiame ir tolimesnių vektorių, dekuoduojame:

$$y = (\underbrace{1\,\underline{0}}_{\underline{0}}\ \underbrace{0\,0}_{0}\ \underbrace{1\,1}_{1}\ \underbrace{1\,0}_{1}\ \underbrace{0\,1}_{0}\ldots)$$

Dekodavome klaidingai. Taigi, dviejų klaidų jau nebeištaisė.



**1.4.1 užduotis.** Pasinagrinėkite, ar ištaisys, pasirinkus didesnį parametrą $t$.

Toliau nagrinėsime tik blokinius kodus.

## 1.5 Kodai, kodavimas ir dekodavimas

### 1.5.1 Kodas ir kodavimas

**1.5.1 apibrėžimas.** Abėcėle *vadinsime baigtinę netuščią aibę.*

Tarkime, turime abėcėlę $A$. Jos elementų skaičių paprastai žymėsime $q$. Jei $q = 2$, tai paprastai laikysime, kad $A = \{0, 1\}$, ir tokią abėcėlę vadinsime *dvinare.*

Tarkime, $n \geqslant 1$ yra sveikasis skaičius. Kaip įprasta, $A^n$ žymėsime aibę visų $n$ ilgio vektorių, kurių koordinatės priklauso aibei $A$:

$$A^n = \{(z_1, \ldots, z_n) \mid z_i \in A,\ i = 1, \ldots, n\}.$$

**1.5.2 apibrėžimas.** *Aibės $A^n$ poaibį $C$ vadinsime* blokiniu kodu virš $A$ *arba tiesiog* kodu. *Vektorius, priklausančius kodui, vadinsime* kodo žodžiais. *Parametrą $n$ vadinsime* kodo ilgiu*, o kodo žodžių skaičių $M$* — kodo dydžiu. *Tokį kodą žymėsime $(n, M)$. Jei $q = 2$, kodą vadinsime* dvejetainiu.

**1.5.3 pavyzdys.** $A = \{0, 1\}$, $n = 5$, $C = \{(0, 0, 0, 0, 0), (1, 1, 0, 0, 0), (0, 1, 1, 1, 1)\}$, $M = 3$. Tai $(5, 3)$ dvejetainis kodas. □

Toliau vektorių žymėjime paprastai praleisime skliaustus ir kablelius, pavyzdžiui, vektorių $(0, 0, 0, 0, 0)$ užrašysime tiesiog 00000.

Tarkime, $C$ yra $(n, M)$ kodas. Tegu $k \geqslant 1$ yra toks sveikasis skaičius, kad $q^k \leqslant M$ (kadangi $M \leqslant q^n$, tai $k \leqslant n$). Laikome, kad informacija, kurią norime persiųsti nepatikimu ryšio kanalu, yra abėcėlės $A$ simbolių seka. Ją skaidome į $k$ ilgio informacijos vektorius $m$. Tarkime, turime kokią nors injekciją $c : A^k \to C$ (tokia injekcija egzistuoja, nes $q^k \leqslant M$). Naudodami šią injekciją, užkoduosime informacijos vektorius $m$, tiksliau, vektorių $m \in A^k$ užkoduosime vektoriumi $x = c(m) \in C$. Paprastai laikysime, kad $c$ yra bijekcija, t. y. $c(A^k) = C$ (tam turi būti $q^k = M$). Tokiu atveju egzistuos atvirkštinė funkcija $c^{-1} : C \to A^k$, kurią galėsime naudoti dekodavimo metu (po to, kai ištaisysime kanale padarytas klaidas, t. y. tada, kai gausime kodo $C$ žodį. Detaliau tai paaiškinta 1.5.3 poskyryje).

**1.5.4 pavyzdys.** Tarkime, $A = \{0, 1\}$, ir turime ilgio $n = 3$ kodą $C = \{000, 101, 011, 110\} \subset A^3$. Tegu $k = 2$, tada $q^k = M = 4$. Apibrėžkime bijekciją $c : A^k \to C$ taip: $c(00) = 000$, $c(01) = 101$, $c(10) = 011$ ir $c(11) = 110$. Tada informacijos srautą skaidome į $k = 2$ ilgio vektorius, ir kiekvieną galimą vektorių koduojame, naudodami bijekciją $c$, pavyzdžiui, informacijos vektorių $m = 01$ koduojame vektoriumi $x = c(01) = 101$. Dekodavimui naudosime atvirkštinę funkciją $c^{-1} : C \to A^k$, kuri yra tokia: $c^{-1}(000) = 00$, $c^{-1}(101) = 01$, $c^{-1}(011) = 10$, $c^{-1}(110) = 11$. □

Matome, kad, jei ilgis $n$ fiksuotas, kodo dydis $M$ turi būti kuo didesnis, kad juo būtų galima užkoduoti kuo daugiau informacijos vektorių (t. y. kuo $M$ didesnis, tuo didesnį $k$, tenkinantį $q^k \leqslant M$, galime parinkti). Tačiau, kuo daugiau žodžių kode, tuo mažiau jie skiriasi vienas nuo kito ir tuo sunkiau nustatyti, kuris vektorius buvo pasiųstas, ir surasti klaidos pozicijas.

Reikia rinktis tokį kodą, kurio ir dydis, ir atstumai tarp kodo žodžių yra kuo didesni. Tai sunkiai suderinami tikslai. Parenkant kodus, ieškoma kompromiso tarp šių dviejų dalykų.

### 1.5.2 Atstumas ir svoris

Dekodavimui labai svarbi yra atstumo tarp vektorių sąvoka. Atstumą tarp vektorių galime matuoti įvairiais būdais. Šiame kurse mes naudosime bene paprasčiausią būdą, pasiūlytą to paties Hemingo, su kurio vardo kodais mes jau susidūrėme.

**1.5.5 apibrėžimas.** *Jei* $x = (x_1, \dots, x_n), y = (y_1, \dots, y_n) \in A^n$, *tai* Hemingo atstumas *tarp vektorių* $x$ *ir* $y$, *žymimas* $d(x, y)$, *yra koordinačių, kuriose jie skiriasi, skaičius:*

$$d(x, y) = |\{i : \ 1 \leqslant i \leqslant n, x_i \neq y_i\}| = \sum_{1 \leqslant i \leqslant n, \ x_i \neq y_i} 1.$$

**1.5.6 pavyzdys.** $A = \{0, 1\}$, $d(0001, 0101) = 1$, $d(0001, 1100) = 3$. □

**1.5.7 pavyzdys.** $A = \{0, 1, 2\}$, $d(1201, 2220) = 3$. □

Įsitikinkite patys, kad Hemingo atstumas yra *metrika*, t. y. $\forall x, y, z \in A^n$ tenkina šias savybes:

1. $d(x, y) \geqslant 0$. Be to, $d(x, y) = 0$ tada ir tik tada, kai $x = y$.
2. $d(x, y) = d(y, x)$.
3. $d(x, y) \leqslant d(x, z) + d(z, y)$ (trikampio nelygybė).

Su atstumu glaudžiai susijusi yra svorio sąvoka, kuri bus ypač naudinga dekoduojant tiesinius kodus.

**1.5.8 apibrėžimas.** *Tarkime, kad abėcėlė* $A$ *yra Abelio grupė sudėties atžvilgiu. Vektoriaus* $x \in A^n$ Hemingo svoris, *žymimas* $w(x)$, *yra jo nenulinių koordinačių skaičius:*

$$w(x) = |\{i : \ 1 \leqslant i \leqslant n, x_i \neq 0\}| = \sum_{1 \leqslant i \leqslant n, \ x_i \neq 0} 1.$$

**1.5.9 pavyzdys.** $A = \{0, 1\}$, $w(0101) = 2$, $w(0000) = 0$. □

**1.5.10 pavyzdys.** $A = \{0, 1, 2\}$, $w(1201) = 3$. □

Nuo šiol laikysime, kad mūsų naudojama abėcėlė $A$ yra Abelio grupė sudėties atžvilgiu.
Nesunku įsitikinti, kad:

1. $w(x) = d(x, 0)$.
2. $d(x, y) = w(x - y) = w(y - x)$.
3. $w(x + y) \leqslant w(x) + w(y)$ (trikampio nelygybės analogas).

Įrodykite šias savybes.
Kadangi šiame kurse naudosime tik Hemingo atstumą ir svorį, tai juos dažniausiai vadinsime tiesiog *atstumu* ir *svoriu*.



### 1.5.3 Dekodavimo taisyklės

Taigi, visą informaciją, kurią norime išsiųsti, skaidome į $k$ ilgio vektorius $m \in A^k$, kiekvieną vektorių užkoduojame $n$ ilgio kodo žodžiu $x = c(m) \in C$ ir siunčiame į kanalą. Iš kanalo gauname iškraipytą vektorių $y \in A^n$, kuris gali nebepriklausyti kodui $C$, t. y. $y$ yra bet kuris vektorius iš aibės $A^n$. Dekodavimo metu vektoriui $y$ priskiriamas vektorius $m' \in A^k$. Dekodavimas paprastai vykdomas dviem etapais. Visų pirma vektoriui $y$ priskiriame kodo žodį $x' \in C$, t. y. ištaisome kanale padarytas klaidas. Antrame etape tiesiog pasinaudojame bijekcijos $c$ atvirkštine funkcija $c^{-1} : C \to A^k$, kad žodžiui $x' \in C$ priskirtume $m' \in A^k$. Pirmame dekodavimo etape naudojama funkcija $f : A^n \to C$ vadinama *dekodavimo taisykle*.

**1.5.11 pavyzdys.** Prisiminkime pakartojimo kodą $R_3$, kuris vektorių $m = 0$ užkoduoja vektoriumi $c = 000$, o $m = 1$ — vektoriumi $c = 111$. Dekodavimas yra toks: kokių simbolių gautame žodyje daugiau, tokiu simboliu ir dekoduojame, pavyzdžiui, jei iš kanalo išėjo vektorius $y = 101$, tai jį dekoduojame 1, nes vektoriuje $y$ yra du vienetai ir tik vienas nulis. Dviem etapais dekodavimą galime užrašyti tokiu būdu: pirmame etape imame artimiausią (matuojant Hemingo atstumu) iš kanalo išėjusiam vektoriui kodo žodį, antrame etape imame informacijos vektorių, atitinkantį tą kodo žodį. Pavyzdžiui, jei iš kanalo išėjo vektorius $y = 101$, tai pirmame etape iš dviejų kodo žodžių 000 ir 111 pasirenkame 111, nes jis arčiau 101, nei 000 (atstumas tarp $y = 101$ ir 000 yra du, tarp $y = 101$ ir 111 yra vienas), ir antrame etape dekoduojame 1, nes jis atitinka 111. □

Tarkime, ryšio kanalu gavome vektorių $y \in A^n$. Koks kodo žodis $x \in C$ buvo išsiųstas? Norėtųsi rasti tokį kodo žodį $x$, kurio išsiuntimo tikimybė yra didžiausia, žinant, kad gavome vektorių $y$. Dekodavimo taisyklė, kai iš kanalo gautą vektorių $y$ dekoduojame kodo žodžiu $x \in C$, maksimizuojančiu tikimybę $P(x|y)$ (taip žymėsime sąlyginę tikimybę, kad į kanalą buvo išsiųstas $x$, jei žinome, kad iš kanalo gautas $y$), vadinama *idealaus stebėtojo taisykle*. T. y. dekoduojame tokiu kodo žodžiu $x$, kad $P(x|y) = \max_{x' \in C} P(x'|y)$.

Be abejo, šią dekodavimo taisyklę sunku naudoti praktikoje, nes tas tikimybes skaičiuoti yra sudėtinga. Nesunkiai galima parodyti, kad esant išpildytoms tam tikroms sąlygoms, ši dekodavimo taisyklė sutampa su *minimalaus atstumo dekodavimo taisykle*: dekoduojame tuo kodo žodžiu, kuris yra artimiausias gautam vektoriui $y$, matuojant Hemingo atstumu, t. y. gavę $y \in A^n$, jį dekoduojame tokiu kodo žodžiu $x \in C$, kad $d(x, y) = \min_{x' \in C} d(x', y)$.

**1.5.12 teorema.** *Tarkime, kad naudojame* $q$*-tainį simetrinį kanalą su iškraipymo tikimybe* $p < \frac{q-1}{q}$, *ir kad visi kodo* $C$ *žodžiai į kanalą perduodami su vienodomis tikimybėmis (vienodai dažnai). Tada idealaus stebėtojo ir minimalaus atstumo dekodavimo taisyklės duoda tą patį rezultatą.*

*Įrodymas.* Tarkime, $x = (x_1, \dots, x_n) \in C$ — į kanalą pasiųstas kodo žodis, $y = (y_1, \dots, y_n) \in A^n$ — iš kanalo išėjęs vektorius. Tarkime, jie skiriasi $u \geqslant 0$ pozicijomis, t. y. $d(x, y) = u$ (tai reiškia, kad kanale padaryta $u$ klaidų).

Teoremą įrodysime, jei parodysime, kad $u$ yra minimalus tada ir tik tada, kai sąlyginė tikimybė $P(x|y)$ yra maksimali.

Pagal sąlyginių tikimybių savybes

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)},$$

kur $P(y|x)$ yra sąlyginė tikimybė, kad iš kanalo išėjo $y$, jei į kanalą įėjo $x$, $P(x)$ — kodo žodžio $x$ išsiuntimo į kanalą tikimybė, $P(y)$ — vektoriaus $y$ gavimo iš kanalo tikimybė. Kadangi visi kodo

$C$ žodžiai į kanalą perduodami su vienodomis tikimybėmis, tai $P(x) = \frac{1}{M}$, kur $M$ — kodo $C$ dydis. Taigi,

$$P(x|y) = \frac{P(y|x)}{MP(y)}.$$

Kadangi $y$ — fiksuotas, tai $P(y)$ taip pat fiksuotas, todėl sąlyginė tikimybė $P(x|y)$ yra maksimali tada ir tik tada, kai sąlyginė tikimybė $P(y|x)$ yra maksimali.

Toliau bandysime rasti ryšį tarp sąlyginės tikimybės $P(y|x)$ ir atstumo $d(x, y) = u$ tarp vektorių $x$ ir $y$.

Naudojamas kanalas yra be atminties, todėl kiekvieno išsiųsto simbolio gavimo tikimybė yra nepriklausoma nuo kitų simbolių tikimybių, taigi

$$P(y|x) = \prod_{i=1}^{n} P(y_i|x_i).$$

Pagal $q$-tainio simetrinio kanalo su iškraipymo tikimybe $p$ savybes, į kanalą įėjus $x_i$, iš kanalo su tikimybe $1-p$ išeina $x_i$ (t. y. klaidos nėra), ir su tikimybe $\frac{p}{q-1}$ išeina kuris nors kitas simbolis (įvyko klaida). Tai yra,

$$P(y_i|x_i) = \begin{cases} 1-p, & \text{jei } y_i = x_i \text{ (klaidos nėra)}, \\ \frac{p}{q-1}, & \text{jei } y_i \neq x_i \text{ (įvyko klaida)}. \end{cases}$$

Kadangi $u = d(x, y)$, tai buvo padaryta lygiai $u$ klaidų, todėl sandaugoje $\prod_{i=1}^{n} P(y_i|x_i)$ lygiai $u$ narių bus lygūs $\frac{p}{q-1}$, o likę $n-u$ nariai bus $1-p$. Todėl

$$P(y|x) = \prod_{i=1}^{n} P(y_i|x_i) = (1-p)^{n-u}\left(\frac{p}{q-1}\right)^u = (1-p)^n\left(\frac{p}{(1-p)(q-1)}\right)^u.$$

Bet $p < \frac{q-1}{q}$, todėl $\frac{p}{(1-p)(q-1)} < 1$. Iš tikro,

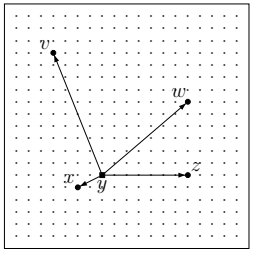
$$\frac{p}{(1-p)(q-1)} < 1 \Leftrightarrow p < (1-p)(q-1) = q - 1 - pq + p \Leftrightarrow pq < q - 1 \Leftrightarrow p < \frac{q-1}{q}.$$

Taigi, kuo $u$ mažesnis, tuo $P(y|x)$ yra didesnė, nes $\frac{p}{(1-p)(q-1)} < 1$. Ką ir reikėjo įrodyti. □

**1.5.13 pastaba.** $p < \frac{q-1}{q} \Leftrightarrow 1 - p > \frac{p}{q-1}$, t. y. teoremos sąlygos reikalavimas ekvivalentus reikalavimui, kad neklaidos tikimybė būtų didesnė už bet kokios klaidos tikimybę.

### 1.5.4 Dekodavimas ir minimalus atstumas

Dekodavimą naudojant minimalaus atstumo dekodavimo taisyklę grafiškai galima pavaizduoti taip:

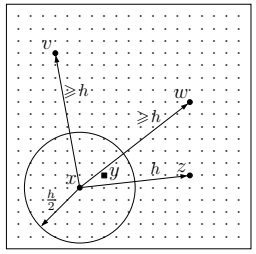





Šiame pavyzdyje kvadratas su taškais yra aibė visų vektorių iš $A^n$, didesni taškai priklauso kodui $C = \{v, w, x, z\} \subseteq A^n$. Iš kanalo gautas vektorius $y$ (žymimas kvadratuku) gali būti bet kuris taškas. Pažiūrime, kuris kodo žodis yra arčiausiai, juo ir dekoduojame (čia rodyklės rodo tai, kad matuojame atstumus tarp $y$ ir kiekvieno kodo žodžio). Šiuo atveju arčiausiai yra kodo žodis $x$, todėl juo ir dekoduojame.

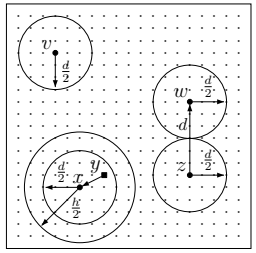
Panagrinėkime toliau kodo $C$ savybes.

Matome, kad kodo žodžiui $x$ artimiausias kitas kodo $C$ žodis yra $z$. Pažymėkime $h$ atstumą tarp $x$ ir $z$. Tada, jei iš kanalo gautas vektorius $y$ yra nutolęs nuo $x$ mažesniu atstumu, nei $h/2$, tai jis tikrai bus dekoduotas žodžiu $x$, nes šis kodo $C$ žodis bus arčiausias. Taigi, apie kodo žodį $x$ galime apibrėžti spindulio $h/2$ apskritimą, ir visi jo viduje esantys taškai bus dekoduojami to apskritimo centru $x$:




Tuo pačiu tai reiškia, kad jei į kanalą buvo pasiųstas kodo žodis $x$, ir kanale buvo padaryta mažiau kaip $h/2$ klaidų, iš kanalo išėjęs žodis $y$ tikrai bus dekoduotas teisingai (visos klaidos bus ištaisytos), nes $y$ bus apskritimo, apibrėžto apie $x$, viduje. Jei klaidų padaryta $h/2$ arba daugiau, dėl teisingo dekodavimo jau nesame garantuoti — kai kada gali dekoduoti teisingai, kai kada klaidingai. Taigi, turime tarsi kodo „gerumo“ („kokybiškumo“) matą, parodantį, kiek kodas tikrai ištaisys klaidų, jei į kanalą buvo pasiųstas kodo žodis $x$.

Tą patį galime padaryti su kiekvienu kodo žodžiu, ir gausime daugybę įvairaus spindulio apskritimų, apibrėžtų apie kodo žodžius. Bet norėtųsi turėti universalų kodo „gerumo“ („kokybiškumo“) matą, nepriklausantį nuo kodo žodžių. Tam iš visų apskritimų spindulių išrenkame mažiausią, tarkim, tai yra $d/2$, apie kiekvieną kodo žodį apibrėžiame spindulio $d/2$ apskritimą, ir tada esame garantuoti, kad jei kanale buvo padaryta mažiau kaip $d/2$ klaidų, iš kanalo išėjęs žodis $y$ tikrai bus dekoduotas teisingai (nepriklausomai nuo to, koks kodo žodis buvo pasiųstas į kanalą), nes jis bus teisingo apskritimo viduje:



Toks $d$ vadinamas kodo minimaliu atstumu. Pateikiame formalų jo apibrėžimą.

**1.5.14 apibrėžimas.** *Tarkime, $C$ yra $(n, M)$ kodas ir $M \geqslant 2$ (kodas $C$ sudarytas iš bent dviejų žodžių). Kodo $C$* minimalus atstumas *$d$ yra mažiausias Hemingo atstumas tarp dviejų skirtingų kodo $C$ žodžių, t. y.*

$$d = \min_{x,y \in C,\, x \neq y} d(x, y).$$

*$(n, M)$ kodą, kurio minimalus atstumas yra $d$, žymėsime $(n, M, d)$.*

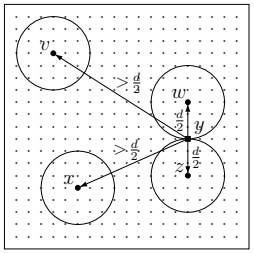
*Tarkime, $M \geqslant 1$. Kodo $C$* minimalus svoris *$w$ yra mažiausias nenulinio kodo $C$ žodžio svoris, t. y.*

$$w = \min_{x \in C,\, x \neq 0} w(x).$$

**1.5.15 pavyzdys.** Tarkime, $A = \{0, 1\}$, $C = \{00001, 11101, 00110\}$. Atstumai tarp kodo žodžių yra tokie: $d(00001, 11101) = 3$, $d(00001, 00110) = 3$ ir $d(11101, 00110) = 4$. Mažiausias atstumas yra trys, todėl kodo $C$ minimalus atstumas $d = 3$. Taigi, kodas $C$ yra $(5, 3, 3)$ kodas.

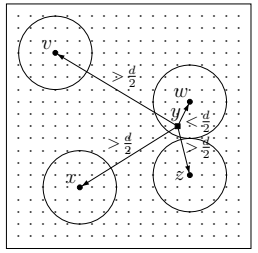
Kodo $C$ nenulinių žodžių svoriai yra tokie: $w(00001) = 1$, $w(11101) = 4$ ir $w(00110) = 2$. Todėl kodo $C$ minimalus svoris yra 1. □

Tarkime, į kanalą pasiuntėme žodį $x$, ir kanale jame buvo padaryta $t$ klaidų. Iš kanalo išėjo vektorius $y$, esantis atstumu $t$ nuo $x$. Jei $t < d/2$, tai $y$ bus kodo žodžio $x$ apskritimo viduje, todėl dekoduosime žodžiu $x$, t. y. ištaisysime kanalo padarytas klaidas. Jei $t = d/2$, tai $y$ gali priklausyti dviem apskritimams, ir tokiu atveju nežinosime, kurio apskritimo centru dekoduoti. Tokia situacija, kai į kanalą buvo pasiųstas kodo žodis $z$, kanale buvo padaryta $d/2$ klaidų, ir iš kanalo išėjęs vektorius $y$ yra vienodai nutolęs nuo kodo žodžių $z$ ir $w$, yra pavaizduota šiame brėžinyje:

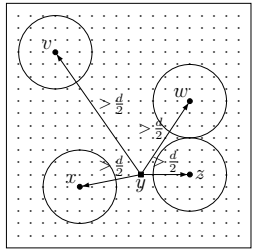



Jei $t > d/2$, tai $y$ gali atsidurti jau kito kodo žodžio skritulyje, ir tokiu atveju bus dekoduotas neteisingai. Pavyzdžiui, šiame brėžinyje pavaizduota, kad, į kanalą pasiuntus kodo žodį $z$, kanale buvo padaryta daugiau nei $d/2$ klaidų, ir iš kanalo išėjęs vektorius pateko jau į kito kodo žodžio $w$ apskritimo vidų:






Jei iš kanalo išėjęs vektorius $y$ neatsiduria jokio kodo žodžio skritulyje, tai, jį dekoduodami artimiausiu kodo žodžiu, galime dekoduoti teisingai, o galime ir klaidingai. Pavyzdžiui, šiame brėžinyje pavaizduota, kad, į kanalą pasiuntus kodo žodį $z$, kanale buvo padaryta daugiau nei $d/2$ klaidų, bet arčiausias iš kanalo išėjusiam žodžiui $y$ visgi išlieka $z$, ir dekoduota bus teisingai:




Taigi, jei klaidų skaičius $t \geqslant d/2$, nesame garantuoti dėl dekodavimo teisingumo.

**1.5.16 apibrėžimas.** *Jei, naudojant minimalaus atstumo dekodavimo taisyklę, dekoduojama visada teisingai, kai siųstame žodyje yra ne daugiau kaip $t$ klaidų, tai kodą $C$ vadiname $t$* klaidų taisančiu kodu.

*$t$ klaidų taisantį kodą vadiname* tiksliai $t$ klaidų taisančiu kodu*, jei jis nėra $t+1$ klaidą taisantis kodas.*

Taigi, kodas $C$ bus tiksliai $t$ klaidų taisantis kodas, jei $t$ yra didžiausias toks skaičius, kad kodas $C$ yra $t$ klaidų taisantis kodas.

**1.5.17 pavyzdys.** 1.5.15 pavyzdžio kodas $C$ yra tiksliai 1 klaidą taisantis kodas. Iš tikrųjų, nesunku patikrinti, kad kanale įvykus vienai klaidai, artimiausias vis tiek liks siųstas kodo žodis, todėl, naudojant minimalaus atstumo dekodavimo taisyklę, dekoduojama bus teisingai (patikrinkite patys!). O įvykus dviems klaidoms, dekoduoti gali ir klaidingai. Pavyzdžiui, jei į kanalą buvo pasiųstas kodo žodis $c = 00001$, kanale įvyko dvi klaidos, iš kanalo išėjo $y = 11001$, tai artimesnis vektoriui $y$ bus kitas kodo žodis $c' = 11101$, todėl, naudojant minimalaus atstumo dekodavimo taisyklę, dekoduojama bus klaidingai. □

Tegu $[a]$ žymi skaičiaus $a$ sveikąją dalį, t. y. didžiausią sveikąjį skaičių, mažesnį arba lygų $a$.

**1.5.18 teorema.** *Tegu $d$ yra kodo $C$ minimalus atstumas. Kodas $C$ yra tiksliai $[(d-1)/2]$ klaidų taisantis kodas.*

*Įrodymas.* Kaip matėme, kodas $C$ yra $t$ klaidų taisantis kodas, jei $t < d/2$. Dėl to jis yra tiksliai $t$ klaidų taisantis kodas, jei $t$ yra didžiausias sveikas skaičius, mažesnis už $d/2$. Jei $d$ yra lyginis, tai didžiausias sveikas skaičius, mažesnis už $d/2$, yra

$$d/2 - 1 = (d-2)/2 = [(d-1)/2].$$

Jei $d$ yra nelyginis, tai didžiausias sveikas skaičius, mažesnis už $d/2$, yra

$$d/2 - 1/2 = (d-1)/2 = [(d-1)/2].$$

Abiem atvejais gauname pageidaujamą rezultatą. □

Todėl stengiamasi sudaryti tokius kodus, kurių minimalus atstumas $d$ būtų kuo didesnis, kad kodas ištaisytų kuo daugiau klaidų.

**1.5.19 pavyzdys.** 1.5.15 pavyzdžio kodo $C$ minimalus atstumas $d = 3$, todėl pagal teoremą jis yra tiksliai $[(d-1)/2] = 1$ klaidą taisantis kodas. □

Kartais svarbu ne tik tai, kiek klaidų kodas ištaiso, bet ir kiek jų aptinka. Kadangi į kanalą siunčiame tik kodo žodžius, tai, jei iš kanalo išeina ne kodo žodis, reiškia, buvo padaryta klaidų. Todėl $t$ klaidų aptinkantis kodas gali būti apibrėžiamas taip:

**1.5.20 apibrėžimas.** *Kodą $C$ vadiname $t$* klaidų aptinkančiu kodu*, jei bet kuriame kodo žodyje įvykus ne daugiau kaip $t$ klaidų, gautas vektorius nepriklauso kodui $C$.*

*$t$ klaidų aptinkantį kodą vadiname* tiksliai $t$ klaidų aptinkančiu kodu*, jei jis nėra $t+1$ klaidą aptinkantis kodas.*

**1.5.21 pavyzdys.** 1.5.15 pavyzdžio kodas $C$ yra tiksliai 2 klaidas aptinkantis kodas. Iš tikrųjų, nesunku patikrinti, kad kanale įvykus vienai ar dviem klaidoms, gautas vektorius nepriklausys kodui $C$. O įvykus trims klaidoms, iš kodo žodžio $c = 00001$ galime gauti kitą kodo žodį $c' = 11101$, todėl kodas $C$ nėra tris klaidas aptinkantis kodas. □

Jei įvyko $d$ iškraipymų, tai gali būti, kad gavome kitą kodo žodį, todėl kodas $C$ yra $t$ klaidų aptinkantis kodas, jei $t < d$. Gauname tokį rezultatą.

**1.5.22 teorema.** *Tegu $d$ yra kodo $C$ minimalus atstumas. Kodas $C$ yra tiksliai $d-1$ klaidą aptinkantis kodas.*

**1.5.23 pavyzdys.** 1.5.15 pavyzdžio kodo $C$ minimalus atstumas $d = 3$, todėl pagal teoremą jis yra tiksliai $d - 1 = 2$ klaidas aptinkantis kodas. □

Taigi, jei norime sukonstruoti gerą kodą esant fiksuotam ilgiui, tai stengiamės, kad jo ir dydis, ir minimalus atstumas būtų kuo didesni. Tai sunkiai suderinami tikslai, todėl ieškoma kompromiso.



## 1.6 Ekvivalentūs kodai

Kai užkoduojame pranešimą ir siunčiame kodo žodį kanalu be atminties, tai galimų iškraipymų požiūriu nėra skirtumo, kuria tvarka persiunčiame to kodo žodžio simbolius (kad ir kuria tvarka juos siųsime, galimi tie patys iškraipymai su tomis pačiomis tikimybėmis). Taigi, šiuo požiūriu kodai, kurie skiriasi tik simbolių eilės tvarka, iš esmės nesiskiria. Tokius kodus vadinsime *ekvivalenčiais*.

Kad galėtume duoti formalų ekvivalenčių kodų apibrėžimą, prisiminkime perstatos (keitinio) sąvoką.

Tarkime, turime baigtinę aibę $I = \{1, \dots, n\}$. Abipusiškai vienareikšmį atvaizdį (bijekciją) $\sigma : I \to I$ vadina *perstata* (arba *keitiniu*). Pavyzdžiui, jei $n = 5$, tai $\sigma : I \to I$, apibrėžta taip: $\sigma(1) = 2$, $\sigma(2) = 3$, $\sigma(3) = 4$, $\sigma(4) = 5$, $\sigma(5) = 1$, yra perstata.

Perstata dažnai užrašoma tokiu pavidalu:

$$\sigma = \begin{pmatrix} 1 & \dots & n \\ \sigma(1) & \dots & \sigma(n) \end{pmatrix}.$$

Pavyzdžiui, paskutinio pavyzdžio perstata galėtų būti užrašyta taip:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}.$$

Perstatas galime naudoti vektorių koordinačių keitimui vietomis. Laikysime, kad $I = \{1, \dots, n\}$ — vektoriaus $x = (x_1, \dots, x_n)$ indeksų aibė. Tada sukeičiame vektoriaus $x$ koordinates taip, kaip nurodo perstata $\sigma$, t. y. pirmą koordinatę $x_1$ perkeliame į $\sigma(1)$ vietą, antrą — į $\sigma(2)$ vietą ir t.t. Gautą vektorių žymėsime $\sigma(x)$. Pavyzdžiui, jei $x = (x_1, x_2, x_3, x_4, x_5)$, tai panaudoję paskutinio pavyzdžio perstatą $\sigma$, gausime vektorių $\sigma(x) = (x_5, x_1, x_2, x_3, x_4)$. Iš tiesų, pirmą koordinatę $x_1$ perkeliame į antrą poziciją ir t.t., o pirmoje vietoje atsiduria $x_5$, nes $\sigma(5) = 1$. Nesunku pastebėti, kad jei $x = (x_1, \dots, x_n)$, tai

$$\sigma(x) = (x_{\sigma^{-1}(1)}, \dots, x_{\sigma^{-1}(n)}), \tag{1.4}$$

kur $\sigma^{-1}$ — atvirkštinė $\sigma$ perstata, t. y. tokia perstata, kad $\sigma^{-1}(\sigma(i)) = i \ \forall i$. Pavyzdžiui, paskutinio pavyzdžio perstatos $\sigma$ atvirkštinė perstata yra tokia perstata:

$$\sigma^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix}.$$

Taigi, perstata $\sigma : I \to I$ apibrėžia funkciją $\sigma : A^n \to A^n$, užduotą (1.4) lygybe. Ši funkcija tiesiog sukeičia vektoriaus koordinates vietomis.

Jei $C$ yra kodas, žymėsime $\sigma(C) = \{\sigma(x) : x \in C\}$, t. y. $\sigma(C)$ bus kodas, gautas sukeitus visų kodo $C$ žodžių koordinates pagal tą pačią perstatą $\sigma$.

**1.6.1 pavyzdys.** Tegu $C = \{0000, 1100, 0011, 1111\}$ yra dvejetainis kodas, perstata

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}.$$

Tada $\sigma(C) = \{0000, 0110, 1001, 1111\}$. □

**1.6.2 apibrėžimas.** *Du ilgio $n$ kodai $C$ ir $C'$ vadinami* ekvivalenčiais*, jei egzistuoja tokia indeksų aibės $\{1, \dots, n\}$ perstata $\sigma$, kad $\sigma(C) = C'$.*

**1.6.3 pavyzdys.** Dvejetainiai kodai $C = \{0000, 1100, 0011, 1111\}$ ir $C' = \{0000, 0110, 1001, 1111\}$ yra ekvivalentūs, nes $C' = \sigma(C)$, kur $\sigma$ yra perstata iš 1.6.1 pavyzdžio. □

Nesunku įsitikinti, kad ekvivalenčių kodų ilgiai, dydžiai, minimalūs atstumai, žodžių svorių pasiskirstymai sutampa (nes sukeitus vektoriaus koordinates vietomis, vektoriaus ilgis bei svoris išlieka tokie patys).

**1.6.4 pavyzdys.** Nustatykime, ar dvejetainiai kodai

$$C = \{1010, 1101, 0011, 1001\} \quad \text{ir} \quad C' = \{1010, 1100, 0110, 1011\}$$

yra ekvivalentūs, ir jei yra, raskime tokią perstatą $\sigma$, kad $C' = \sigma(C)$.

Pastebėsime, kad šį uždavinį galima išspręsti išsamiosios paieškos metodu, patikrinant visas galimas perstatas. Bet ilgio $n$ perstatų yra $n!$, todėl jei $n$ yra didelis, šis metodas nėra efektyvus. Tačiau yra sukurta įvairių taisyklių, padedančių gerokai sumažinti galimų perstatų skaičių. Pasinaudosime paprasčiausiomis iš jų.

Visų pirma panagrinėkime abiejų kodų žodžius. Matome, kad abiejų kodų žodžių svorių pasiskirstymai sutampa: kodai turi po tris svorio 2 žodžius ir po vieną svorio 3 žodį. Jei nesutaptų, iškart galėtume tvirtinti, kad kodai neekvivalentūs. Kadangi turi tik po vieną svorio 3 žodį, tai aišku, jog jei tokia perstata $\sigma$ egzistuoja, tai jinai žodį 1101 atvaizduoja į 1011. Iš čia iškart gauname, kad jei $\sigma$ egzistuoja, tai $\sigma(3) = 2$, nes nulis iš trečios žodžio 1101 pozicijos perkeliamas į antrą žodžio 1011 poziciją. Taigi, gavome šiek tiek informacijos apie $\sigma$.

Toliau galime suskaičiuoti, kiek vienetų stovi pirmo kodo žodžių kiekvienoje pozicijoje, ir palyginti su antro kodo vienetų skaičiumi. Kad aiškiau matytųsi, užrašykime kodo žodžius stulpeliu:

$$C = \begin{Bmatrix} 1010 \\ 1101 \\ 0011 \\ 1001 \end{Bmatrix} \quad \text{ir} \quad C' = \begin{Bmatrix} 1010 \\ 1100 \\ 0110 \\ 1011 \end{Bmatrix}.$$

Perstatant kodo žodžių koordinates, šie stulpeliai yra perkeliami į kitą vietą, taigi, vienetų skaičius juose išlieka toks pat. Kodo $C$ pirmame stulpelyje yra 3 vienetai, antrame — 1, trečiame — 2, ketvirtame — 3. Kodo $C'$ atitinkamai 3,2,3,1. Vėlgi, matome, pasiskirstymas sutampa (abiem atvejais turime po vieną stulpelį su 1 vienetu, po vieną su 2 vienetais, ir po du su 3 vienetais) — jei nesutaptų, kodai nebūtų ekvivalentūs. Matome, kad antras kodo $C$ stulpelis, kuriame yra 1 vienetas, būtinai keliauja į ketvirtą poziciją (t. y. jei kodai ekvivalentūs ir perstata $\sigma$ egzistuoja, tai $\sigma(2) = 4$), ir trečias — į antrą (tą mes jau matėme, palyginę kodo žodžių svorius).

Taigi, jau žinome dvi perstatos $\sigma$ reikšmes: $\sigma(2) = 4$ ir $\sigma(3) = 2$. Yra tik dvi perstatos su tokiomis reikšmėmis:

$$\sigma' = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 2 & 3 \end{pmatrix} \quad \text{ir} \quad \sigma'' = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix}$$

Patikriname jas ir gauname, kad jos abi tinka, t. y. abi jos kodo $C$ koordinates sukeičia taip, kad gauname kodą $C'$. Iš tikro,

$$\sigma'(C) = \{1100, 1011, 0110, 1010\} = C' \quad \text{ir} \quad \sigma''(C) = \{0110, 1011, 1100, 1010\} = C'.$$

Taigi, kodai ekvivalentūs. □



# 2 skyrius
# Tiesiniai kodai

Bendru atveju kodą apibrėžėme kaip vektorių aibę ir nagrinėjome jo savybes. Šiame skyriuje kodui suteiksime tiesinės erdvės struktūrą. Matysime, kad tai palengvins ir kodavimą, ir dekodavimą, ir kodo savybių tyrimą.

## 2.1 Kai kurių algebrinių struktūrų priminimas

1.3.5 poskyryje prisiminėme operacijos ir grupės sąvokas. Šiame skyriuje tęsime priminimą kai kurių sąvokų, jums žinomų iš algebros kurso, kurių reikia tiesinių kodų apibrėžimui ir jų savybių tyrimui.

### 2.1.1 Baigtiniai kūnai

Šiame poskyryje prisiminsime baigtinių kūnų (dar vadinamų Galua kūnais, angl. *finite fields, Galois fields*) pagrindines sąvokas, nes ateityje nagrinėsime kodus, apibrėžtus virš baigtinės abėcėlės, turinčios kūno struktūrą.

Visų pirma prisiminkime *kūno* (kartais dar vadinamo *lauku*) apibrėžimą.

**2.1.1 apibrėžimas.** Kūnas $(A, +, \cdot)$ — *tai aibė $A$, sudaryta bent iš dviejų elementų, kurioje apibrėžtos sudėties „+“ ir daugybos „$\cdot$“ operacijos, tenkinančios tokias savybes.*

1. *$(A, +)$ yra komutatyvioji grupė. Prisiminkime, kad* 0 *žymi neutralų sudėties atžvilgiu elementą (*nulį*).*
2. *$(A \setminus \{0\}, \cdot)$ yra grupė.*
3. *Galioja distributyvumo dėsniai: visiems $a, b, c \in A$*

$$a \cdot (b + c) = a \cdot b + a \cdot c,$$
$$(a + b) \cdot c = a \cdot c + b \cdot c.$$

**2.1.2 pastaba.** Prisiminkime, kad neutralų daugybos atžvilgiu elementą žymime „1“ ir vadiname *vienetu*. Pagal kūno apibrėžimą gauname, kad kiekviename kūne yra bent du elementai — 0 ir 1.

**2.1.3 pastaba.** Galima parodyti, kad jei $A$ yra kūnas, tai $0 \cdot a = 0 \ \ \forall a \in A$.

**2.1.4 pavyzdys.** Jei $p$ — pirminis skaičius, tai sveikųjų skaičių moduliu $p$ aibė sudaro baigtinį kūną iš $p$ elementų

$$\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\},$$

vadinamą *pirminiu kūnu*. Jame operacijos atliekamos moduliu $p$ (t. y. atlikus veiksmą dalinama iš $p$ ir imama liekana). Pavyzdžiui, kūne $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$ gauname, kad $3 + 4 = 2$, nes $3 + 4 = 7 \equiv 2 \bmod 5$. Taip pat ir $3 \cdot 4 = 2$, nes $3 \cdot 4 = 12 \equiv 2 \bmod 5$. *Dvinariame kūne* $\mathbb{F}_2 = \{0, 1\}$ gauname, kad $1 + 1 = 0$. □

**2.1.5 apibrėžimas.** *Kūną, turintį baigtinį skaičių elementų, vadinsime* baigtiniu kūnu. *Baigtinį kūną, turintį $q$ elementų, žymėsime $\mathbb{F}_q$ (angliškoj literatūroj jis dažnai žymimas $GF(q)$).*

**2.1.6 pavyzdys.** Pirminis kūnas $\mathbb{F}_p$ yra baigtinis. □

Baigtinis kūnas iš $q$ elementų egzistuoja tada ir tik tada, kai $q = p^m$, kur $p$ — pirminis, $m \geqslant 1$. Visi baigtiniai kūnai, turintys $q$ elementų, yra izomorfiški, todėl galima laikyti, kad toks kūnas yra vienintelis.

Tarkime, $q = p^m$, kur $p$ — pirminis, $m \geqslant 1$. Jei $m = 1$, tai $q = p$ ir $\mathbb{F}_q$ yra pirminis kūnas, t. y. sveikųjų skaičių moduliu $q$ aibė. Jei $m > 1$, tai $\mathbb{F}_q$ nėra sveikųjų skaičių moduliu $q$ aibė, t. y. veiksmai su kūno $\mathbb{F}_q$ elementais atliekami kitaip, nei sveikųjų skaičių moduliu $q$ aibėje. Tuo pačiu gauname, kad jei $r$ nėra pirminis, sveikųjų skaičių moduliu $r$ aibė nėra kūnas.

Visi baigtiniai kūnai yra komutatyvūs, t. y. daugyba yra komutatyvi.

**2.1.7 apibrėžimas.** *Mažiausias toks sveikas skaičius $s$, kad $\underbrace{1 + \cdots + 1}_{s \text{ kartų}} = 0$, kur $0$ ir $1$ yra atitinkamai kūno $K$ nulis ir vienetas, vadinamas kūno $K$* charakteristika. *Jei toks skaičius neegzistuoja (pavyzdžiui, realiųjų skaičių kūne), kūno charakteristika laikoma lygia nuliui.*

**2.1.8 pavyzdys.** Pirminio kūno $\mathbb{F}_p$ charakteristika yra $p$, nes

$$\underbrace{1 + \cdots + 1}_{p \text{ kartų}} = p \equiv 0 \pmod{p},$$

ir $p$ yra mažiausias skaičius, tenkinantis šią savybę. □

Baigtinio kūno $\mathbb{F}_{p^m}$ charakteristika yra $p$.

Jei $p$ yra kūno $K$ charakteristika, tai

$$p\beta = 0 \text{ visiems } \beta \in K. \tag{2.1}$$

Pavyzdžiui, $2\beta = 0$ visiems $\beta \in \mathbb{F}_2$.

Iš (2.1) lygybės gauname, kad

$$(\beta + \gamma)^p = \beta^p + \gamma^p \ \forall \beta, \gamma \in \mathbb{F}_{p^m}. \tag{2.2}$$

Pagal indukciją taip pat gauname

$$(\beta + \gamma)^{p^i} = \beta^{p^i} + \gamma^{p^i} \ \forall \beta, \gamma \in \mathbb{F}_{p^m} \ \forall i \geqslant 1. \tag{2.3}$$

## 2.1.2 Tiesinė erdvė

Tegu $A$ — kūnas, $n \geqslant 1$ — sveikasis skaičius. Aibė $V \subset A^n$ vadinama *tiesine erdve virš $A$*, jei

$$v, u \in V \Rightarrow av + bu \in V \ \forall a, b \in A.$$

**2.1.9 pavyzdys.** Pagal apibrėžimą nesunku patikrinti, kad $V = \{000, 011, 101, 110\}$ yra tiesinė erdvė virš $\mathbb{F}_2$. □



**2.1.10 pastabos.** 1. Nulinis vektorius visada priklauso tiesinei erdvei $V$, nes pagal apibrėžimą bet kuri erdvės $V$ vektorių $u$ ir $v$ tiesinė kombinacija priklauso $V$, tuo pačiu ir $0 \cdot u + 0 \cdot v = 0 \in V$.

2. Dvinariu atveju tiesinės erdvės apibrėžimo sąlyga tampa

$$v, u \in V \Rightarrow v + u \in V. \tag{2.4}$$

Iš tikrųjų, dvinariu atveju $a$ ir $b$ gali įgyti tik dvi reikšmes — 0 ir 1, todėl tėra keturios galimos tiesinės kombinacijos:

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

Antra ir trečia priklauso $V$ pagal sąlygą, todėl lieka patikrinti, ar $0 \in V$ ir ar teisinga (2.4) sąlyga. Bet pastebėkime, kad sąlyga $0 \in V$ išplaukia iš (2.4) sąlygos, nes iš pastarosios gauname, kad $0 = u + u \in V$. Todėl lieka tik (2.4) sąlyga.

Tiesinės erdvės $V$ vektorių rinkinys $u_1, \ldots, u_s$ vadinamas *tiesiškai nepriklausomu*, jei

$$(a_1u_1 + \cdots + a_su_s = 0, a_i \in A \ \forall i) \Rightarrow (a_1 = \cdots = a_s = 0).$$

Šį apibrėžimą galima suformuluoti ir iš kitos pusės: tiesinės erdvės $V$ vektorių rinkinys $u_1, \ldots, u_s$ vadinamas tiesiškai nepriklausomu, jei bet kokia tiesinė jų kombinacija su bent vienu nenuliniu koeficientu yra nenulinis vektorius:

$$(a_i \in A \ \forall i \ \text{ ir } \ \exists i : a_i \neq 0, 1 \leqslant i \leqslant s) \Rightarrow (a_1u_1 + \cdots + a_su_s \neq 0).$$

**2.1.11 užduotis.** Įrodykite, kad galioja šios savybės.

1. Rinkinys iš vieno vektoriaus yra tiesiškai priklausomas tada ir tik tada, kai tas vektorius yra nulinis vektorius.
2. Jei nulinis vektorius priklauso vektorių rinkiniui, tai tas vektorių rinkinys yra tiesiškai priklausomas.
3. Vektorių rinkinys yra tiesiškai priklausomas tada ir tik tada, kai kuris nors to rinkinio vektorius yra likusių vektorių tiesinė kombinacija.
4. Dviejų nenulinių vektorių $u, v$ rinkinys yra tiesiškai priklausomas tada ir tik tada, kai egzistuoja $a \in A$ toks, kad $u = av$.
5. Dviejų dvejetainių nenulinių vektorių $u, v$ rinkinys yra tiesiškai priklausomas tada ir tik tada, kai tie vektoriai lygūs.
6. Jei vektorių aibė yra tiesiškai nepriklausoma, tai bet koks netuščias jos poaibis irgi yra tiesiškai nepriklausomas.

Tiesinės erdvės $V$ tiesiškai nepriklausomų vektorių rinkinys $u_1, \ldots, u_s$ vadinamas erdvės $V$ *baze*, jei kiekvieną erdvės $V$ vektorių galima išreikšti vektorių $u_1, \ldots, u_s$ tiesine kombinacija, t. y.

$$\forall v \in V \ \exists a_1, \ldots, a_s \in A : v = a_1u_1 + \cdots + a_su_s.$$

**2.1.12 pavyzdys.** Tiesinės erdvės iš 2.1.9 pavyzdžio bazė yra $\{011, 101\}$. Be to, aibės $\{011, 110\}$ ir $\{101, 110\}$ taip pat yra šios erdvės bazės. □

**2.1.13 teorema.** *Tarkime, $V \subset A^n$ yra tiesinė erdvė. Tada $V$ turi bazę, sudarytą iš ne daugiau kaip $n$ vektorių. Jei ji sudaryta iš $s$ vektorių, tai:*

1. *Bet koks vektorių iš $V$ rinkinys, sudarytas iš $s+1$ vektoriaus, yra tiesiškai priklausomas.*
2. *Kiekviena $V$ bazė yra sudaryta iš $s$ vektorių.*
3. *Bet kurie $s$ tiesiškai nepriklausomi $V$ vektoriai sudaro $V$ bazę.*
4. *Kiekvienas tiesinės erdvės $V$ vektorius* vienareikšmiškai *išreiškiamas bazės vektorių tiesine kombinacija.*

Erdvės $V$ *dimensija*, žymima $\dim V$, yra bazės vektorių skaičius $s$.

### 2.1.3 Baigtiniai kūnai detaliau

Pirminis[1] kūnas $\mathbb{F}_p = \{0, 1, 2, \ldots, p-1\}$ yra kūno $\mathbb{F}_{p^m}$ poaibis. Kūnas $\mathbb{F}_{p^m}$ yra tiesinė erdvė virš $\mathbb{F}_p$, $\dim \mathbb{F}_{p^m} = m$. Jei $\beta_0, \beta_1, \cdots, \beta_{m-1}$ yra erdvės $\mathbb{F}_{p^m}$ bazė virš $\mathbb{F}_p$, tai

$$\mathbb{F}_{p^m} = \{a_0\beta_0 + a_1\beta_1 + \cdots + a_{m-1}\beta_{m-1} \mid a_i \in \mathbb{F}_p \ \forall i\}.$$

Aibė $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$ yra ciklinė multiplikacinė grupė, t. y.

$$\exists \alpha \in \mathbb{F}_q^* : \mathbb{F}_q^* = \{1, \alpha, \alpha^2, \cdots, \alpha^{q-2}\}, \alpha^{q-1} = 1.$$

Toks $\alpha$ yra vadinamas *primityviu kūno $\mathbb{F}_q$ elementu* (jis nebūtinai yra vienintelis). Ši baigtinio kūno elementų išraiška leidžia lengvai dauginti ir dalinti: dauginant (dalinant) du ciklinės grupės elementus pakanka sudėti (atimti) jų laipsnių rodiklius (moduliu $q-1$).

Jei $\alpha \in \mathbb{F}_{p^m}$ yra primityvus elementas, tai $\{1, \alpha, \alpha^2, \cdots, \alpha^{m-1}\}$ yra kūno $\mathbb{F}_{p^m}$ bazė virš $\mathbb{F}_p$, todėl

$$\mathbb{F}_{p^m} = \{a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1} \mid a_i \in \mathbb{F}_p \ \forall i\}.$$

Ši baigtinio kūno elementų išraiška leidžia lengvai sudėti ir atimti: sudedame (atimame) panariui koeficientus prie $\alpha$ laipsnių. Koeficientus sudėti (atimti) mokame, nes jie priklauso kūnui $\mathbb{F}_p$. Praktikoje į elementą $a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}$ patogu žiūrėti kaip į vektorių $(a_0, a_1, \ldots, a_{m-1}) \in \mathbb{F}_p^m$, ir veiksmus atlikti su vektoriais.

Jei galėtume nesunkiai pereiti nuo vienos baigtinio kūno elementų išraiškos prie kitos, tai galėtume elementus ir sudėti (atimti), ir dauginti (dalinti). Netrukus pamatysime, kaip tai padaryti. Bet prieš tai — dar viena baigtinio kūno elementų išraiška.

Jei $f(x)$ yra pirminis $m$-tojo laipsnio polinomas virš $\mathbb{F}_p$, tai $\mathbb{F}_{p^m}$ yra izomorfiškas faktoržiedžiui $\mathbb{F}_p[x]/(f(x))$ (prisiminkime, kad faktoržiedis $\mathbb{F}_p[x]/(f(x))$ yra polinomų virš $\mathbb{F}_p$ dalybos iš $f(x)$ liekanų aibė, kurioje veiksmai atliekami moduliu $f(x)$). Taigi, galime laikyti, kad

$$\mathbb{F}_{p^m} = \{g(x) = g_0 + g_1x + \cdots + g_{m-1}x^{m-1} \mid g_i \in \mathbb{F}_p \ \forall i\},$$

ir veiksmai atliekami $\bmod f(x)$.

[1] Tekstas smulkesniu šriftu yra neprivalomas.



Visiems $p$ — pirminiams ir $m \geqslant 1$ egzistuoja toks pirminis $m$-tojo laipsnio polinomas $f(x) \in \mathbb{F}_p[x]$, kurio šaknis yra kūno $\mathbb{F}_{p^m}$ primityvus elementas $\alpha$, t. y. $f(\alpha) = 0$. Toks polinomas vadinamas *primityviu polinomu* (jis nebūtinai yra vienintelis).

Primityvus polinomas $f(x)$ leidžia susieti dvi baigtinio kūno $\mathbb{F}_{p^m}$ elementų išraiškas, tiksliau, leidžia primityvaus elemento laipsnius $\alpha^m, \alpha^{m+1}, \ldots, \alpha^{p^m-2}$ išreikšti bazės $1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{m-1}$ vektorių tiesine kombinacija. Iš tikro, tegu $f(x) = f_0 + f_1x + \cdots + f_mx^m$, $f_i \in \mathbb{F}_p$ $\forall i$. Kadangi $f(\alpha) = 0$, tai $f_0 + f_1\alpha + \cdots + f_m\alpha^m = 0$. Iš čia gauname, kad $\alpha^m = -\frac{1}{f_m}(f_0 + f_1\alpha + \cdots + f_{m-1}\alpha^{m-1})$ — išreiškėme bazės vektorių tiesine kombinacija. Aukštesnius $\alpha$ laipsnius išreiškiame taip pat, pavyzdžiui, $\alpha^{m+1} = \alpha^m\alpha$, įstatome gautą $\alpha^m$ išraišką ir t.t.

Nors primityvių polinomų yra ne vienas ir visi jie gali būti naudojami veiksmams su baigtinio kūno elementais, yra tam tikri „standartiniai“ primityvūs polinomai, kurie paprastai ir yra naudojami. 2.1 lentelėje pateikiame kelis jų pavyzdžius mažiems $p$ ir $m$.

| $p$ | Primityvūs polinomai | $p$ | Primityvūs polinomai |
|---|---|---|---|
| 2 | $x+1$ | 3 | $x^4+x^3+2$ |
| | $x^2+x+1$ | | $x^5+x^4+x^2+1$ |
| | $x^3+x+1$ | | $x^6+x^5+2$ |
| | $x^4+x+1$ | 5 | $x^2+x+2$ |
| | $x^5+x^2+1$ | | $x^3+x^2+2$ |
| | $x^6+x+1$ | | $x^4+x^3+x+3$ |
| | $x^7+x+1$ | 7 | $x^2+x+3$ |
| | $x^8+x^4+x^3+x^2+1$ | | $x^3+x^2+x+2$ |
| | $x^9+x^4+1$ | 11 | $x^2+x+7$ |
| | $x^{10}+x^3+1$ | 13 | $x^2+x+2$ |
| 3 | $x+1$ | 17 | $x^2+x+10$ |
| | $x^2+x+2$ | 19 | $x^2+x+2$ |
| | $x^3+2x^2+1$ | 23 | $x^2+22x+19$ |

2.1 lentelė: Primityvūs polinomai

**2.1.14 pavyzdys.** Sudarykime kūną $\mathbb{F}_8$. Matome, kad $p = 2$, $m = 3$ (nes $8 = 2^3$). Žinome, kad $\mathbb{F}_8^* = \{1, \alpha, \alpha^2, \cdots, \alpha^6\}$, $\alpha^7 = 1$, kur $\alpha$ yra kūno $\mathbb{F}_8$ primityvus elementas. Tai leidžia dauginti kūno $\mathbb{F}_8$ elementus, pavyzdžiui, $\alpha^2\alpha^3 = \alpha^5$, $\alpha^3\alpha^6 = \alpha^9 = \alpha^7\alpha^2 = \alpha^2$ (kad $\alpha^9 = \alpha^2$, galėjome pasakyti iš karto, nes rodiklių skaičiavimai vyksta moduliu $q-1$, t. y. mod7). Be abejo, $0 \cdot \alpha^i = 0$ $\forall i$.

Be to, žinome, kad $\{1, \alpha, \alpha^2\}$ yra kūno $\mathbb{F}_8$ bazė virš $\mathbb{F}_2$, todėl

$$\begin{aligned} \mathbb{F}_8 &= \{a_0 + a_1\alpha + a_2\alpha^2 \mid a_i \in \mathbb{F}_2 \ \forall i\} \\ &= \{0, 1, \alpha, 1+\alpha, \alpha^2, 1+\alpha^2, \alpha+\alpha^2, 1+\alpha+\alpha^2\}. \end{aligned}$$

Tai leidžia lengvai sudėti ir atimti, pavyzdžiui, $(1+\alpha)+(\alpha+\alpha^2) = 1+2\alpha+\alpha^2 = 1+\alpha^2$ (nes kūno $\mathbb{F}_8$ charakteristika yra $p = 2$, t. y. $1+1 = 2 = 0$ ir $-1 = 1$).

Kad galėtume pereiti nuo vienos išraiškos prie kitos, pasinaudosime primityviu polinomu iš 2.1 lentelės. Matome, kad, kai $p = 2$ ir $m = 3$, galime naudoti primityvų polinomą $f(x) = x^3+x+1$. Primityvus elementas $\alpha$ yra jo šaknis, t. y. $f(\alpha) = 0$, todėl $\alpha^3 + \alpha + 1 = 0$. Iš čia $\alpha^3 = -\alpha - 1 = \alpha + 1$ — išreiškėme bazės vektorių tiesine kombinacija.

Tada

$$\begin{aligned} \alpha^4 &= \alpha^3\alpha = (\alpha+1)\alpha = \alpha^2 + \alpha, \\ \alpha^5 &= \alpha^4\alpha = (\alpha^2+\alpha)\alpha = \alpha^3 + \alpha^2 = \alpha + 1 + \alpha^2, \\ \alpha^6 &= \alpha^5\alpha = (\alpha+1+\alpha^2)\alpha = \alpha^2 + \alpha + \alpha^3 = \alpha^2 + \alpha + \alpha + 1 = \alpha^2 + 1. \end{aligned}$$

Žinome, kad $\alpha^7 = 1$. Patikrinkime, ar iš tikrųjų:

$$\alpha^7 = \alpha^6\alpha = (\alpha^2+1)\alpha = \alpha^3 + \alpha = \alpha + 1 + \alpha = 1.$$

| | | |
|---|---|---|
| 0 | 0 | 000 |
| 1 | 1 | 100 |
| $\alpha$ | $\alpha$ | 010 |
| $\alpha^2$ | $\alpha^2$ | 001 |
| $\alpha^3$ | $1+\alpha$ | 110 |
| $\alpha^4$ | $\alpha+\alpha^2$ | 011 |
| $\alpha^5$ | $1+\alpha+\alpha^2$ | 111 |
| $\alpha^6$ | $1+\alpha^2$ | 101 |

2.2 lentelė: Kūno $\mathbb{F}_8$ elementai

Taigi, dabar, atlikdami veiksmus su kūno $\mathbb{F}_8$ elementais, lengvai galime pereiti nuo vienos išraiškos prie kitos. Pavyzdžiui, $\alpha^4+\alpha^6=(\alpha^2+\alpha)+(\alpha^2+1)=\alpha+1=\alpha^3$.

Kaip buvo pastebėta, į elementą $a_0+a_1\alpha+a_2\alpha^2$ patogu žiūrėti kaip į vektorių $(a_0,a_1,a_2)\in\mathbb{F}_2^3$, ir veiksmus atlikti su vektoriais. Pavyzdžiui, elementą $\alpha^2+\alpha$ atitinka vektorius 011. Programuojant veiksmus su nedideliais baigtiniais kūnais, patogu pasidaryti lentelę, kuri susietų primityvaus elemento $\alpha$ laipsnius su tokiais vektoriais. Tada du elementai bus sudauginami, sudedant jų laipsnių rodiklius, o norint sudėti du elementus, lentelėje bus surandami juos atitinkantys vektoriai, sudedami (moduliu $p$), o rezultatas vėl paverčiamas $\alpha$ laipsniu. Pavyzdžiui, $\alpha^4+\alpha^6\to 011+101=110\to\alpha^3$.

Reziumuodami 2.2 lentelėje pateikiame kūno $\mathbb{F}_8$ elementų išraiškas. Pirmame stulpelyje yra primityvaus elemento $\alpha$ laipsnis, antrame — jo išraiška bazės $\{1,\alpha,\alpha^2\}$ vektorių tiesine kombinacija, ir trečiame — atitinkamas vektorius iš $\mathbb{F}_2^3$. □

**2.1.15 pavyzdys.** Panagrinėkime kūną $\mathbb{F}_9$. Kadangi $9=3^2$, tai $p=3$ ir $m=2$. Taigi, $\mathbb{F}_3=\{0,1,2\}$ yra kūno $\mathbb{F}_9$ poaibis ($2+2=4=1$, $3=0$, $-1=2$, $-2=1$). Taip pat žinome, kad $\mathbb{F}_9^*=\{1,\alpha,\alpha^2,\cdots,\alpha^7\}, \alpha^8=1$, kur $\alpha$ yra kūno $\mathbb{F}_9$ primityvus elementas. Be to, $\{1,\alpha\}$ yra kūno $\mathbb{F}_9$ bazė virš $\mathbb{F}_3$, todėl

$$\begin{aligned}\mathbb{F}_9 &= \{a_0+a_1\alpha \mid a_i\in\mathbb{F}_3\ \forall i\}\\ &= \{0,1,2,\alpha,\alpha+1,\alpha+2,2\alpha,2\alpha+1,2\alpha+2\}.\end{aligned}$$

Primityvus polinomas (iš 2.1 lentelės) būtų $f(x)=x^2+x+2$, taigi, $\alpha^2+\alpha+2=0$ ir $\alpha^2=-\alpha-2=2\alpha+1$. Toliau

$$\begin{aligned}\alpha^3 &= \alpha^2\alpha=(2\alpha+1)\alpha=2\alpha^2+\alpha=2(2\alpha+1)+\alpha=4\alpha+2+\alpha=2\alpha+2,\\ \alpha^4 &= \alpha^3\alpha=(2\alpha+2)\alpha=2\alpha^2+2\alpha=2(2\alpha+1)+2\alpha=4\alpha+2+2\alpha=2,\\ \alpha^5 &= \alpha^4\alpha=2\alpha,\\ \alpha^6 &= \alpha^5\alpha=2\alpha^2=2(2\alpha+1)=\alpha+2,\\ \alpha^7 &= \alpha^6\alpha=(\alpha+2)\alpha=\alpha^2+2\alpha=2\alpha+1+2\alpha=\alpha+1.\end{aligned}$$

Dabar galime atlikti veiksmus su kūno $\mathbb{F}_9$ elementais. Pavyzdžiui, $\alpha^3\alpha^6=\alpha^9=\alpha$, $(\alpha^2)^{-1}=\alpha^{-2}=\alpha^6$ (nes rodiklių veiksmai atliekami moduliu 8), $\alpha^6-\alpha^3=(\alpha+2)-(2\alpha+2)=\alpha+2-2\alpha-2=-\alpha=2\alpha=\alpha^5$, ir pan.

2.3 lentelėje pateikiame kūno $\mathbb{F}_9$ elementus. Pirmame stulpelyje yra primityvaus elemento $\alpha$ laipsnis, antrame — jo išraiška bazės $\{1,\alpha\}$ vektorių tiesine kombinacija, ir trečiame — atitinkamas vektorius iš $\mathbb{F}_3^2$. □

## 2.2 Tiesinio kodo apibrėžimas

Pradedame nagrinėti didelę kodų šeimą, vadinamą tiesiniais kodais. Praktiškai visos svarbesnės kodų šeimos yra tiesinių kodų šeimos pošeimiai. Šiame skyriuje nagrinėsime savybes, bendras visiems tiesiniams kodams, o vėliau pereisime prie atskirų tiesinių kodų pošeimių.

Nuo šiol abėcėlė bus $A=\mathbb{F}_q$, $q=p$, $p$ — pirminis[2], t. y. abėcėlė bus pirminis kūnas $\mathbb{F}_p$. Tegu $n\geq 1$.

[2]Visa čia išdėstyta teorija tinka ir nepirminiams kūnams. Toliau yra pateikti ir keli (neprivalomi) pavyzdžiai virš nepirminių kūnų.



| | | |
|---|---|---|
| 0 | 0 | 00 |
| 1 | 1 | 10 |
| $\alpha$ | $\alpha$ | 01 |
| $\alpha^2$ | $1+2\alpha$ | 12 |
| $\alpha^3$ | $2+2\alpha$ | 22 |
| $\alpha^4$ | 2 | 20 |
| $\alpha^5$ | $2\alpha$ | 02 |
| $\alpha^6$ | $2+\alpha$ | 21 |
| $\alpha^7$ | $1+\alpha$ | 11 |

2.3 lentelė: Kūno $\mathbb{F}_9$ elementai

**2.2.1 apibrėžimas.** *Kodą $C\subset\mathbb{F}_q^n$ vadinsime* tiesiniu*, jei $C$ yra tiesinė erdvė. Tiesinio kodo dimensijos $k$ ir ilgio $n$ santykį $\frac{k}{n}$ vadinsime tiesinio kodo* koeficientu.

Jei tiesinio kodo $C$ ilgis yra $n$, dimensija $k$, minimalus atstumas $d$, tai visas kodas žymimas $C[n,k,d]$ arba tiesiog $[n,k,d]$. Jei minimalus atstumas nesvarbus arba nežinomas, žymima $C[n,k]$ arba $[n,k]$.

Taigi, tiesinis kodas yra tiesiog tiesinė erdvė. Todėl galime pasinaudoti visomis tiesinių erdvių savybėmis. Pavyzdžiui, žinome, kad tiesinė erdvė turi bazę.

**2.2.2 pavyzdys.** Imkime tiesinį erdvės $\mathbb{F}_3^4$ poerdvį $C$, generuotą vektorių 2102 ir 1120. Tai reiškia, kad šie vektoriai sudaro poerdvio bazę, o visi kiti išreiškiami jų tiesinėmis kombinacijomis:

$$\begin{aligned}0\cdot 2102+0\cdot 1120 &= 0000,\\ 0\cdot 2102+1\cdot 1120 &= 1120,\\ 0\cdot 2102+2\cdot 1120 &= 2210,\\ 1\cdot 2102+0\cdot 1120 &= 2102,\\ 1\cdot 2102+1\cdot 1120 &= 0222,\\ 1\cdot 2102+2\cdot 1120 &= 1012,\\ 2\cdot 2102+0\cdot 1120 &= 1201,\\ 2\cdot 2102+1\cdot 1120 &= 2021,\\ 2\cdot 2102+2\cdot 1120 &= 0111.\end{aligned}$$

Taigi, tiesinis kodas $C=\{0000,1120,2210,2102,0222,1012,1201,2021,0111\}$. □

**2.2.3 teorema.** *1. Tiesinio $[n,k]$ kodo virš $\mathbb{F}_q$ dydis yra $q^k$.*

*2. Tiesinio kodo minimalus atstumas yra lygus minimaliam svoriui.*

Teoremos įrodymas paliekamas skaitytojui kaip užduotis.

**2.2.4 pavyzdys.** 2.2.2 pavyzdžio kodas yra $[4,2]$ kodas virš $\mathbb{F}_3$, todėl pagal teoremos pirmą dalį, jo dydis turi būti $3^2=9$. Tą mes ir matėme. □

Pagal teoremos antrą dalį, tiesinio kodo minimalus atstumas randamas žymiai paprasčiau, negu netiesinio kodo. Užtenka surasti minimalų svorį, kuris ir bus minimalus atstumas.

**2.2.5 pavyzdys.** Kodo iš 2.2.2 pavyzdžio minimalų atstumą rasime taip. Randame jo minimalų svorį: mažiausio svorio nenulinis žodis yra svorio 3 žodis (tiesą pasakius, visų šio kodo nenulinių žodžių svoris yra 3). Todėl šio kodo minimalus atstumas irgi yra trys. □

## 2.3 Generuojanti matrica

Tegu $C[n,k]$ yra tiesinis kodas virš $\mathbb{F}_q$. Kad jis būtų visiškai apibrėžtas, nebūtina išrašyti visus $q^k$ jo žodžius. Pakanka nurodyti jo bazę. Kodavimo teorijoje dažnai patogiau bazę užrašyti ne kaip vektorių rinkinį, o kaip matricą.

**2.3.1 apibrėžimas.** *Tiesinio kodo $C[n,k]$ virš $\mathbb{F}_q$* generuojančia matrica *vadiname $k \times n$ matricą virš $\mathbb{F}_q$, kurios eilutės sudaro kodo $C$ bazę.*

**2.3.2 pavyzdys.** 2.2.2 pavyzdžio kodo viena iš generuojančių matricų yra

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

**2.3.3 pastabos.**

1. Generuojančios matricos eilučių skaičius lygus kodo dimensijai.
2. Tiesinis kodas gali turėti kelias generuojančias matricas, nes jis gali turėti kelias skirtingas bazes, o ir tos pačios bazės vektorius išrikiavę kita tvarka, gausime kitą generuojančią matricą.

Jei turime kokio nors tiesinio kodo $C[n,k]$ virš $\mathbb{F}_q$ generuojančią matricą $G$, tai visus kodo $C$ žodžius ir tiktai juos gausime imdami generuojančios matricos $G$ eilučių tiesines kombinacijas. Nesunku pastebėti, kad matricos $G$ eilučių $G_1, \dots, G_k$ tiesinė kombinacija su koeficientais $x_1, \dots, x_k$ yra lygi vektoriaus-eilutės $x = (x_1, \dots, x_k)$ ir matricos $G$ sandaugai $xG$, t. y. $(x_1, \dots, x_k)G = x_1G_1 + \dots + x_kG_k$.

**2.3.4 pavyzdys.**

$$(x_1, x_2)\begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} = (2x_1 + x_2, x_1 + x_2, 2x_2, 2x_1) = x_1(2102) + x_2(1120). \qquad \square$$

$\square$

Todėl $C = \{xG \mid x \in \mathbb{F}_q^k\}$.

Atvaizdis $x \mapsto xG$ apibrėžia abipusiškai vienareikšmę erdvės $\mathbb{F}_q^k$ ir tiesinio kodo $C$ žodžių atitiktį. Todėl šį atvaizdį galima interpretuoti kaip šaltinio informacijos, pateikiamos erdvės $\mathbb{F}_q^k$ žodžiais, kodavimą kodo $C$ žodžiais.

Taigi, kodavimas tiesiniu kodu visai paprastas — ilgio $k$ informacijos vektorių $x$ virš $\mathbb{F}_q$ dauginame iš kodo $C$ generuojančios matricos $G$ ir gauname užkoduotą žodį $xG \in C$.

**2.3.5 pavyzdys.** Kodavimui naudojame dvejetainį tiesinį $[4,3]$ kodą, kurio generuojanti matrica yra

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

Tarkime, šaltinis perduoda tokį dvejetainį srautą: $110\ 010\ 001\ 111\ 101\ 010\cdots$ (čia patogumo dėlei srautą iškart suskaidome ilgio $k = 3$ vektoriais). Tada užkoduota seka, kurią siųsime kanalu, bus tokia: $1011\ 0111\ 1010\ 0001\ 0110\ 0111\cdots$ $\square$



Kadangi generuojančios matricos $G$ eilutės yra tiesiškai nepriklausomos, tai jos rangas yra lygus eilučių skaičiui $k$. Iš algebros žinome, kad tada iš matricos galime išrinkti tokius $k$ stulpelių, iš kurių sudarytos kvadratinės matricos determinantas būtų nelygus 0. Tarkime, $s_1, s_2, \dots, s_k$ yra tokių stulpelių numeriai, išrikiuoti didėjančia tvarka.

Tada

1) sukeisdami matricos $G$ eilutes vietomis,
2) daugindami jas iš nenulinių kūno $\mathbb{F}_q$ elementų,
3) pridėdami prie kurios nors $G$ eilutės kitą $G$ eilutę, padaugintą iš kūno $\mathbb{F}_q$ elemento,

galime gauti tokio pavidalo matricą:

$$G' = \begin{pmatrix} \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots \\ \cdots & 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots \\ & \vdots & & \vdots & & \vdots & \ddots & \vdots & \\ \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & \cdots \end{pmatrix} \tag{2.5}$$

Čia vienetinės matricos stulpelius gauname būtent $s_1, s_2, \dots, s_k$ pozicijose. Gautoji matrica taip pat yra kodo $C$ generuojanti matrica, nes atliekant išvardintas matricų operacijas jos rangas nesumažėja ir jos eilutės išlieka kode $C$.

**2.3.6 pastaba.** Atkreipkite dėmesį, kad veiksmus galima atlikti tik su generuojančios matricos eilutėmis, jokiu būdu ne su stulpeliais.

**2.3.7 pavyzdys.** Pertvarkykime 2.3.2 pavyzdžio generuojančią matricą.

$$\begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \begin{matrix} \\ \cdot 2 \\ \cdot 1 \downarrow \end{matrix} \sim \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 2 & 2 \end{pmatrix} \begin{matrix} \\ \cdot 2 \\ \cdot 2 \uparrow \end{matrix} \sim \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Kad gautume pirmame stulpelyje pirmą vienetinės matricos stulpelį, pirmos matricos pirmą eilutę padauginome iš 2, o taip pat pridėjome prie antros. Gauname antrą matricą. Tada antrą antros matricos eilutę padauginome iš 2, o taip pat padauginę iš 2 pridėjome prie pirmos, kad gautume antrame stulpelyje antrą vienetinės matricos stulpelį (žr. trečią matricą). Taigi, vienetinę matricą gauname pirmame ir antrame stulpeliuose, t. y. šiuo atveju $k = 2$, o $s_1 = 1$ ir $s_2 = 2$. $\square$

**2.3.8 teorema.** *Du tiesiniai $[n,k]$ kodai virš $\mathbb{F}_q$ yra lygūs tada ir tik tada, kai jų generuojančias matricas galima suvesti į tą pačią (2.5) pavidalo matricą su tais pačiais stulpeliais $s_1, \dots, s_k$.*

Įrodykite šį teiginį savarankiškai.

Tai būdas patikrinti, ar, esant duotoms dviems generuojančioms matricoms, jos generuoja tą patį kodą ar ne. Užtenka pasirinkti $s_1, \dots, s_k$, suvesti matricas į (2.5) lygybės pavidalą ir patikrinti, ar gauname tą pačią matricą.

**2.3.9 pavyzdys.** Tegu

$$G = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \quad \text{ir} \quad G' = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}.$$

2.3.7 pavyzdyje matėme, kad matricoje $G$ vienetinę matricą galima gauti pirmame ir antrame stulpeliuose:

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

Pabandykime ir matricos $G'$ pirmame ir antrame stulpeliuose gauti vienetinę matricą.

$$G' = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix} \begin{matrix} \cdot 2 \\ \\ \end{matrix} \begin{matrix} \cdot 1 \\ \downarrow \end{matrix} \sim \begin{pmatrix} 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{matrix} \cdot 1 \\ \\ \end{matrix} \begin{matrix} \cdot 2 \\ \uparrow \end{matrix} \sim \begin{pmatrix} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Gavome tą pačią matricą. Taigi, matricos $G$ ir $G'$ generuoja tą patį tiesinį kodą. □

**2.3.10 apibrėžimas.** *Matricą*

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots \\ 0 & 1 & \cdots & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \cdots \\ 0 & 0 & \cdots & 1 & \cdots \end{pmatrix}$$

*vadinsime* standartinio pavidalo *matrica.*

Tai $k \times n$ matrica ($k \leqslant n$), kurios pirmuose $k$ stulpelių stovi vienetinė matrica, o likusi matricos dalis yra bet kokia. Standartinio pavidalo matricą paprastai žymėsime taip: $G = (I|A)$, čia $I$ — vienetinė $k \times k$ matrica, $A$ — kokia nors $k \times (n-k)$ matrica.

**2.3.11 teorema.** *Bet kuris tiesinis kodas yra ekvivalentus tiesiniam kodui, turinčiam standartinio pavidalo generuojančią matricą.*

*Įrodymas.* Kaip matėme, bet kurią generuojančią matricą galime suvesti į (2.5) lygybės pavidalą. Paskui sukeičiame gautos matricos stulpelius taip, kad kiekvienam $i$ $s_i$-tasis stulpelis atsidurtų $i$-tojoje pozicijoje, kad gautume standartinio pavidalo matricą. Gauta standartinio pavidalo matrica jau galbūt nebebus pradinio kodo generuojanti matrica, bet jinai bus ekvivalentaus kodo generuojanti matrica, nes jinai generuos kodą, kuris nuo pradinio skirsis tik koordinačių perstata. □

Ši teorema tvirtina, kad paprastai užtenka nagrinėti kodus, turinčius standartinio pavidalo generuojančią matricą, nes visi kiti kodai yra jiems ekvivalentūs.

**2.3.12 pavyzdys.** Turime tiesinį $[5, 3]$ kodą $C$ virš $\mathbb{F}_q$, generuotą matricos $G$. Raskime kodą, ekvivalentų kodui $C$, turintį standartinio pavidalo generuojančią matricą.

1. Tegu $q = 5$, t. y. $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$, čia skaičiuojama moduliu 5 (taigi, $-1 = 4$, $5 = 0$, $6 = 1$ ir t.t.). Tegu

$$G = \begin{pmatrix} 4 & 3 & 1 & 4 & 3 \\ 3 & 1 & 2 & 0 & 4 \\ 4 & 1 & 4 & 2 & 4 \end{pmatrix}.$$



Bandome suvesti $G$ į standartinį pavidalą. Bandydami ir rasime tiesiškai nepriklausomų stulpelių numerius $s_1, s_2, s_3$. Naudosime standartines matricų eilučių operacijas, išvardintas 47 puslapyje prieš (2.5) lygybę. Trumpai prisiminkime, kaip jos taikomos. Taigi, pirmą matricos $G$ stulpelį norime perdaryti į pirmą vienetinės matricos stulpelį, t. y. pirmoje koordinatėje norime gauti 1, kitur — 0.

Kad vietoj 4 gautume 1, pirmą eilutę turime padalinti iš 4, arba, kitaip sakant, padauginti iš $4^{-1}$. Bet kas tai yra elemento 4 atvirkštinis elementas $4^{-1}$? Tai toks elementas, kurį padauginę iš 4, gauname 1 (prisiminkime, kad visos operacijos atliekamos moduliu 5). Nesunkiai įsitikiname, kad $4^{-1} = 4$, nes $4 \cdot 4 = 16 \equiv 1 \pmod 5$. Taigi, pirmą eilutę dauginame iš 4.

Toliau, antroje pirmo stulpelio pozicijoje reikia vietoj 3 gauti 0. Tam pirmą eilutę, padaugintą iš kažkokio skaičiaus $h$, pridėsime prie antros. Kam lygus $h$? Jį galime rasti iš lygybės $4h + 3 = 0$ (pirmos eilutės pirmą elementą dauginame iš $h$, pridedame prie antros eilutės pirmo elemento, ir gauname 0), t. y. $h = (-3)4^{-1} = 2 \cdot 4 = 8 = 3$. Be abejo, $h$ galėjome rasti ir kitaip: iš pradžių pirmą eilutę padaliname iš 4, kad gautume pirmoje pozicijoje 1, o tada padauginame iš $-3$. Tada aišku, kad pridėję pirmą eilutę prie antros, pirmoje pozicijoje gausime 0, nes sudėsime $-3$ iš pirmos eilutės su 3 iš antros. Bet ir tokiu būdu gausime tą patį $h$, nes jei daliname iš 4 ir dauginame iš $-3$, tai $h = 4^{-1}(-3) = 3$. Taigi, pirmą eilutę padauginame iš 3 ir pridedame prie antros. Lygiai taip pat apskaičiuojame, iš ko reikia padauginti pirmą eilutę, kad pridėję ją prie trečios, gautume 0 pirmoje pozicijoje. Gauname:

$$G = \begin{pmatrix} 4 & 3 & 1 & 4 & 3 \\ 3 & 1 & 2 & 0 & 4 \\ 4 & 1 & 4 & 2 & 4 \end{pmatrix} \begin{matrix} \cdot 4 \\ \\ \\ \end{matrix} \begin{matrix} \cdot 3 \\ \downarrow \\ \\ \end{matrix} \begin{matrix} \cdot 4 \\ \\ \downarrow \end{matrix} \sim \begin{pmatrix} 1 & 2 & 4 & 1 & 2 \\ 0 & 0 & 0 & 2 & 3 \\ 0 & 3 & 3 & 3 & 1 \end{pmatrix}.$$

Dabar antrame stulpelyje norime gauti antrą vienetinės matricos stulpelį, t. y. atitinkamai 0, 1 ir 0. Bet antro stulpelio antroje pozicijoje dabar stovi 0 — kad ir iš ko daugintume antrą eilutę, vistiek negausime 1. Tokiu atveju ieškome eilutės, kurios antroje pozicijoje yra ne nulis. Viršuje (virš antros eilutės) ieškoti negalime, nes ten jau sutvarkyta (jau pirmos eilutės pirmoje pozicijoje stovi 1, todėl pirmos eilutės perkelti kitur negalime), galime ieškoti tik apačioje (po antra eilute). Matome, kad trečia eilutė tinka, todėl sukeičiame antrą ir trečią eilutes vietomis:

$$G \sim \begin{pmatrix} 1 & 2 & 4 & 1 & 2 \\ 0 & 0 & 0 & 2 & 3 \\ 0 & 3 & 3 & 3 & 1 \end{pmatrix} \begin{matrix} \\ \updownarrow \\ \\ \end{matrix} \sim \begin{pmatrix} 1 & 2 & 4 & 1 & 2 \\ 0 & 3 & 3 & 3 & 1 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix}.$$

Gauname antrą vienetinės matricos stulpelį:

$$G \sim \begin{pmatrix} 1 & 2 & 4 & 1 & 2 \\ 0 & 3 & 3 & 3 & 1 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix} \begin{matrix} \cdot 2 \\ \\ \\ \end{matrix} \begin{matrix} \cdot 1 \\ \uparrow \\ \\ \end{matrix} \begin{matrix} \cdot 0 \\ \\ \downarrow \end{matrix} \sim \begin{pmatrix} 1 & 0 & 2 & 4 & 3 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix}.$$

Vėl ta pati problema — trečio stulpelio trečioje pozicijoje nepavyks padaryti 1, ir ieškoti eilutės, su kuria būtų galima sukeisti trečią eilutę, nebėra kur (ieškoti galime tik apačioje, t. y. po trečia eilute). Tai reiškia, kad pirmi trys stulpeliai yra tiesiškai priklausomi, ir juose padaryti vienetinės matricos nepavyks. Nieko tokio, kažkur vistiek pavyks, tai renkamės kitą stulpelį. Imkime kitą stulpelį — einant iš eilės, ketvirtą, — ir bandykime jame gauti

vienetinės matricos trečią stulpelį. Tai visai nesunku:

$$G \sim \begin{pmatrix} 1 & 0 & 2 & 4 & 3 \\ 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 2 & 3 \end{pmatrix} \begin{matrix} \\ \\ \cdot 3 \end{matrix} \begin{matrix} \\ \\ \cdot 2 \end{matrix} \begin{matrix} \\ \\ \cdot 3 \end{matrix} \sim \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix}.$$

Gavome (2.5) lygybės pavidalo matricą su vienetinės matricos stulpeliais 1, 2 ir 4 pozicijose. Sukėlę šiuos stulpelius į pradžią (t. y. sukeitę trečią ir ketvirtą stulpelius vietomis), gauname standartinio pavidalo generuojančią matricą

$$G' = \begin{pmatrix} 1 & 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 1 & 3 \\ 0 & 0 & 1 & 0 & 4 \end{pmatrix},$$

kuri nėra kodo $C$ generuojanti matrica (kaip matėme, kodas $C$ neturi standartinio pavidalo generuojančios matricos, nes jo generuojančios matricos pirmi trys stulpeliai yra tiesiškai priklausomi), bet generuoja ekvivalentų kodą $C'$, kuris nuo kodo $C$ skiriasi tik tuo, kad jame trečia ir ketvirta koordinatė yra sukeistos vietomis.

2. Tegu $q = 9$. Skaičiavimams naudosime kūno $\mathbb{F}_9$ lentelę iš 45 puslapio (2.3 lentelė, 2.1.15 pavyzdys).

   Tegu

$$G = \begin{pmatrix} \alpha^2 & \alpha & \alpha^6 & \alpha^4 & \alpha^3 \\ 1 & \alpha^2 & \alpha^5 & \alpha^3 & \alpha \\ \alpha^6 & \alpha^2 & \alpha^3 & \alpha^4 & 1 \end{pmatrix}$$

   ir $x = (\alpha^2, \alpha^4, \alpha)$. Lygiai taip pat, kaip pereitame pavyzdyje apskaičiuojame, iš ko reikia dauginti pirmą eilutę, kad gautume pirmą vienetinės matricos stulpelį. Pavyzdžiui, kad vietoj $\alpha^2$ gautume 1, pirmą eilutę dauginame iš $\left(\alpha^2\right)^{-1} = \alpha^{-2} = \alpha^6$. Kad gautume 0 vietoj 1 antros eilutės pirmoje pozicijoje, pirmą eilutę padauginame iš tokio $h$, kad $\alpha^2 h + 1 = 0$, t. y. $h = (-1)\left(\alpha^2\right)^{-1} = 2\alpha^{-2} = \alpha^4\alpha^6 = \alpha^1 0 = \alpha^2$ (galima buvo skaičiuoti ir taip: $2\alpha^{-2} = 2\alpha^6 = 2(\alpha + 2) = 2\alpha + 1 = \alpha^2$), ir pridedame prie antros. Taip pat, kad gautume 0 vietoj $\alpha^6$ trečios eilutės pirmoje pozicijoje, pirmą eilutę padauginame iš tokio $h$, kad $\alpha^2 h + \alpha^6 = 0$, t. y. $h = (-\alpha^6)\left(\alpha^2\right)^{-1} = 2\alpha^6\alpha^{-2} = \alpha^4\alpha^6\alpha^6 = \alpha^1 6 = 1$, ir pridedame prie trečios. Ir t.t.

$$\begin{aligned} G &= \begin{pmatrix} \alpha^2 & \alpha & \alpha^6 & \alpha^4 & \alpha^3 \\ 1 & \alpha^2 & \alpha^5 & \alpha^3 & \alpha \\ \alpha^6 & \alpha^2 & \alpha^3 & \alpha^4 & 1 \end{pmatrix} \begin{matrix} \cdot\alpha^6 \\ \\ \\ \end{matrix} \begin{matrix} \cdot\alpha^2 \\ \downarrow \\ \\ \end{matrix} \begin{matrix} \cdot 1 \\ \downarrow \\ \\ \end{matrix} \\ &\sim \begin{pmatrix} 1 & \alpha^7 & \alpha^4 & \alpha^2 & \alpha \\ 0 & \alpha & \alpha^2 & 1 & 0 \\ 0 & 1 & 1 & 1 & \alpha^5 \end{pmatrix} \begin{matrix} \cdot\alpha^7 \\ \\ \\ \end{matrix} \begin{matrix} \cdot\alpha^2 \\ \uparrow \\ \\ \end{matrix} \begin{matrix} \cdot\alpha^3 \\ \downarrow \\ \\ \end{matrix} \\ &\sim \begin{pmatrix} 1 & 0 & 1 & \alpha^6 & \alpha \\ 0 & 1 & \alpha & \alpha^7 & 0 \\ 0 & 0 & \alpha^2 & \alpha^5 & \alpha^5 \end{pmatrix} \begin{matrix} \\ \\ \cdot\alpha^6 \end{matrix} \begin{matrix} \\ \\ \cdot\alpha^3 \end{matrix} \begin{matrix} \\ \uparrow \\ \end{matrix} \begin{matrix} \\ \\ \cdot\alpha^2 \end{matrix} \\ &\sim \begin{pmatrix} 1 & 0 & 0 & \alpha^5 & \alpha^2 \\ 0 & 1 & 0 & \alpha^6 & 1 \\ 0 & 0 & 1 & \alpha^3 & \alpha^3 \end{pmatrix} = G' \end{aligned}$$

   Matome, kad kodas $C$ turi standartinio pavidalo generuojančią matricą. Kadangi kodas visada yra sau ekvivalentus (su tapačiąja perstata $\sigma$ gauname, kad $\sigma(C) = C$), tai gavome ekvivalentų kodą $C'$ (šiuo atveju $C' = C$), turintį standartinio pavidalo generuojančią matricą.

□



Pastebėkime, kad kodavimo procedūra $x \mapsto xG$, kai naudojama standartinio pavidalo generuojanti matrica $G = (I|A)$, yra paprastesnė. Iš tikro, jei $x = (x_1, \ldots, x_k)$, tai

$$xG = x(I|A) = (xI|xA) = (x|xA) = (x_1, \ldots, x_k, c_{k+1}, \ldots, c_n),$$

kur $(c_{k+1}, \ldots, c_n) = xA$. Taigi, koduojant prie ilgio $k$ vektoriaus $x$ tiesiog prijungiame $n-k$ kontrolinių simbolių $(c_{k+1}, \ldots, c_n)$, gautų dauginant $x$ iš matricos $A$. Vėliau matysime, kad standartinio pavidalo generuojanti matrica supaprastina ir dekodavimą.

**2.3.13 pavyzdys.** Vektorių $x$ užkoduokime, naudodami 2.3.12 pavyzdyje gautas standartinio pavidalo generuojančias matricas.

1. Tegu $x = (142)$. Koduodami vektorių $x$ kodu $C'$ iš 2.3.12 pavyzdžio pirmos dalies, gauname kodo žodį $xG' = (14212)$. Matome, kad iš tikro užkoduotame vektoriuje pirmos $k = 3$ koordinatės yra iš pradinio vektoriaus $x$.

2. Užkodavę vektorių $x = (\alpha^2, \alpha^4, \alpha)$ kodu $C'$ iš 2.3.12 pavyzdžio antros dalies, gauname $xG' = (\alpha^2, \alpha^4, \alpha, 1, 0)$.

□

**2.3.14 pavyzdys.** Rasti 1.2 poskyryje pateiktų paprastų pavyzdžių generuojančias matricas.

1. *Pakartojimo kodas $R_n$.* Tai dvejetainis kodas. Priminsiu, kad 0 užkoduojame $00\cdots0$, 1 — $11\cdots1$ (pakartojame 0 ar 1 $n$ kartų). Jei šis kodas yra tiesinis, tai jo generuojanti matrica $G$ bus tokia, kad $0 \cdot G = 00\cdots0$ ir $1 \cdot G = 11\cdots1$. Taigi, $G$ bus $1 \times n$ matrica. Nesunku pastebėti, kad $G = (11\cdots1)$.

2. *Kontrolinio simbolio kodas.* Irgi dvejetainis kodas. Pranešimą $x = (x_1, x_2, \ldots, x_k)$ užkoduojame vektoriumi $c = (x_1, x_2, \ldots, x_k, x_{k+1})$, kur $x_{k+1} \equiv \sum_{i=1}^{k} x_i \pmod 2$. Jei užrašytume kūno $\mathbb{F}_2$ operacijomis, tai gautume

$$x_{k+1} = \sum_{i=1}^{k} x_i. \tag{2.6}$$

   Taigi, jei šis kodas yra tiesinis, tai jo generuojanti matrica $G$ tenkins $xG = c$. Matome, kad tai $k \times (k+1)$ matrica. Aišku, kad ji yra standartinio pavidalo, nes pirmos $k$ kodo žodžio $c$ koordinatės lygios vektoriui $x$. Taigi, pirmuose $k$ matricos $G$ stulpelių stovi $k \times k$ vienetinė matrica. Lieka išsiaiškinti, kaip atrodo $(k+1)$-asis matricos $G$ stulpelis. Prisiminę, kaip dauginamas vektorius ir matrica, matome, kad jei matricos $G$ $(k+1)$-asis stulpelis yra $(g_1, g_2, \ldots, g_k)^T$ (čia ir toliau $y^T$ yra transponuotas vektorius ar matrica $y$), tai vektoriaus $xG$ $(k+1)$-oji koordinatė yra lygi $\sum_{i=1}^{k} g_i x_i$. Matome, kad (2.6) lygybėje esančią sumą gauname tada, kai visi $g_i$ yra lygūs 1. Taigi, $(k+1)$-asis matricos $G$ stulpelis yra sudarytas tik iš vienetų. Todėl kontrolinio simbolio kodas yra tiesinis, ir jo generuojanti matrica yra

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

3. *ISBN kodas.* Vektorių $(a_1, a_2, \ldots, a_9)$, sudarytą iš dešimtainių skaitmenų, užkoduojame vektoriumi $(a_1, a_2, \ldots, a_{10})$, kur $a_{10} \equiv \sum_{i=1}^{9} i a_i \pmod{11}$, t. y.

$$a_{10} = \sum_{i=1}^{9} i a_i \tag{2.7}$$

kūne $\mathbb{F}_{11}$. Vėlgi matome, kad matrica $G$ yra standartinio pavidalo, o paskutinis — dešimtas — stulpelis pagal (2.7) lygybę yra $(1, 2, \ldots, 9)^T$. Taigi, matrica bus

$$G = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 9 \end{pmatrix}.$$

4. *Asmens kodo* kodavimui naudojamą matricą galime gauti analogiškai, kaip ir ISBN kodui. Paliekama skaitytojui.

5. *Kodas $[t^2 + 2t, t^2]$, kai $t = 2$.* Tai $[8, 4]$ dvejetainis kodas. Nesunku pastebėti, kad jis vektorių $x = (x_1, x_2, x_3, x_4)$ užkoduoja vektoriumi $c = (x_1, x_2, x_1 + x_2, x_3, x_4, x_3 + x_4, x_1 + x_3, x_2 + x_4)$. Taip pat, kaip anksčiau, randame generuojančios matricos $G$ stulpelius. Pavyzdžiui, kadangi sandaugos $xG$ rezultato $c$ trečia koordinatė yra $x_1 + x_2 = 1 \cdot x_1 + 1 \cdot x_2 + 0 \cdot x_3 + 0 \cdot x_4$, tai trečias matricos $G$ stulpelis bus $(1, 1, 0, 0)^T$. Taigi, kodo generuojanti matrica bus

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

6. $[7, 4]$ *Hemingo kodas.* Tai $[7, 4]$ dvejetainis kodas. Nesunku pastebėti, kad jis vektorių $x = (x_1, x_2, x_3, x_4)$ užkoduoja vektoriumi $c = (x_1, x_2, x_3, x_4, x_1 + x_2 + x_3, x_2 + x_3 + x_4, x_1 + x_3 + x_4)$. Taigi, kodo generuojanti matrica bus

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

□

## 2.4 Dualus kodas ir kontrolinė matrica

### 2.4.1 Dualus kodas

Tegu $x = (x_1, \ldots, x_n)$ ir $y = (y_1, \ldots, y_n)$ yra erdvės $\mathbb{F}_q^n$ vektoriai. Jų *skaliarine sandauga* vadinsime įprastą vektorių skaliarinę sandaugą, t. y.

$$x \cdot y = x_1 y_1 + \cdots + x_n y_n,$$



čia $x \cdot y \in \mathbb{F}_q$, t. y. sandaugos ir sumos operacijos atliekamos kūne $\mathbb{F}_q$. Vektoriai vadinami *ortogonaliais (statmenais)*, jei jų skaliarinė sandauga lygi nuliui. Pavyzdžiui, vektoriai 111 ir 101 yra ortogonalūs virš baigtinio kūno $\mathbb{F}_2$, nes

$$111 \cdot 101 = 1 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 = 1 + 0 + 1 = 0.$$

**2.4.1 apibrėžimas.** *Tegu $C[n, k]$ yra tiesinis kodas virš $\mathbb{F}_q$ . Kodo $C$* dualus kodas*, žymimas $C^\perp$, yra kodo $C$ ortogonali erdvė, t. y. aibė vektorių, ortogonalių kiekvienam kodo $C$ žodžiui:*

$$C^\perp = \{x \in \mathbb{F}_q^n : x \cdot y = 0 \ \forall y \in C\}.$$

**2.4.2 pavyzdys.** Tarkime, dvejetainis tiesinis kodas $C[3, 2]$ yra generuotas matricos

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

Kaip 2.2.2 pavyzdyje galime gauti, kad $C = \{000, 110, 101, 011\}$. Kodo $C$ dualus kodas $C^\perp$ bus sudarytas iš tų erdvės $\mathbb{F}_2^3$ vektorių (jų yra aštuoni: $000, 001, 010, 011, 100, 101, 110, 111$), kurie ortogonalūs kiekvienam kodo $C$ vektoriui. Nesunkiai patikriname, kad tik du vektoriai tenkina šią sąlygą: $C^\perp = \{000, 111\}$. □

**2.4.3 pastaba.** Ortogonalumo sąvoka erdvėse virš baigtinių kūnų skiriasi nuo ortogonalumo sąvokos virš realiųjų skaičių kūno. Pavyzdžiui, erdvėje $\mathbb{R}^n$ tik nulinis vektorius yra ortogonalus pats sau, todėl bet kurios $\mathbb{R}^n$ erdvės ir jos ortogonalios erdvės sankirta visada yra $\{0\}$ (pavyzdžiui, plokštumos ir jai statmenos tiesės). Erdvėse virš baigtinių kūnų ši savybė nebegalioja, pavyzdžiui, vektorius $x = (1, 1, 0, \ldots, 0) \in \mathbb{F}_2^n$ yra ortogonalus pats sau, nes $x \cdot x = 1 + 1 = 0$, todėl gali būti, kad $x \in C$ ir $x \in C^\perp$. Gali netgi būti, kad visi kodo $C$ vektoriai yra ortogonalūs visiems kodo vektoriams, todėl $C \subseteq C^\perp$.

**2.4.4 teorema.** *Tiesinio kodo $C[n, k]$ dualus kodas yra tiesinis $[n, n - k]$ kodas.*

Be įrodymo.

**2.4.5 pavyzdys.** Iš tiesų, 2.4.2 pavyzdžio dualus kodas $C^\perp = \{000, 111\}$ yra tiesinis $[3, 1]$ kodas, jo generuojanti matrica yra $G^\perp = (111)$. □

**2.4.6 teiginys.** *Bet kuriam tiesiniam kodui $C$ turime, kad $(C^\perp)^\perp = C$.*

*Įrodymas.* Visų pirma parodysime, kad $C \subseteq (C^\perp)^\perp$. Tarkime, $x \in C$. Bet kuris kodo $C$ žodis yra ortogonalus bet kuriam kodo $C^\perp$ žodžiui, todėl $x \cdot y = 0 \ \forall y \in C^\perp$. Bet tai savo ruožtu reiškia, kad $x \in (C^\perp)^\perp$. Todėl $C \subseteq (C^\perp)^\perp$.

Iš kitos pusės, ką tik matėme, kad jei $C$ yra $[n, k]$ kodas, tai $C^\perp$ yra tiesinis $[n, n - k]$ kodas. Analogiškai gauname, kad $(C^\perp)^\perp$ yra $[n, n - (n - k)]$, t. y. $[n, k]$ kodas.

Taigi, $C \subseteq (C^\perp)^\perp$ ir $\dim C = \dim (C^\perp)^\perp$, todėl $C = (C^\perp)^\perp$. □

Taigi, įrodėme, kad jei $C^\perp = D$, tai $D^\perp = C$. Todėl visus tiesinius kodus galima suskirstyti į poras, kur kiekvienoje poroje kodai yra vienas kitam dualūs.

**2.4.7 pavyzdys.** 2.4.2 pavyzdyje matėme, kad kodo $C = \{000, 110, 101, 011\}$ dualus kodas yra $C^\perp = \{000, 111\}$. Lygiai taip pat nesunkiai galime įsitikinti, kad kodo $C^\perp$ dualus kodas yra $(C^\perp)^\perp = \{000, 110, 101, 011\} = C$. □

### 2.4.2 Ekvivalenčių kodų dualūs kodai

Ekvivalenčių kodų dualūs kodai taip pat ekvivalentūs. Ir netgi daugiau, jie ekvivalentūs su ta pačia perstata. Tiksliau, galioja toks teiginys.

**2.4.8 teiginys.** *Jei $C$ ir $C'$ yra ekvivalentūs kodai, ir $\sigma$ yra tokia perstata, kad $C' = \sigma(C)$, tai $C'^{\perp} = \sigma(C^{\perp})$.*

*Įrodymas.*

$$\begin{aligned} & x \in C'^{\perp} \\ \iff \quad & x \cdot y = 0 \ \forall y \in C' = \sigma(C) && (2.8) \\ \iff \quad & x \cdot \sigma(c) = 0 \ \forall c \in C && (2.9) \\ \iff \quad & \sigma^{-1}(x) \cdot c = 0 \ \forall c \in C && (2.10) \\ \iff \quad & \sigma^{-1}(x) \in C^{\perp} \\ \iff \quad & x \in \sigma(C^{\perp}). \end{aligned}$$

Ką ir reikėjo įrodyti. Pakomentuosime įrodymą. (2.8) ir (2.9) yra ekvivalentūs, nes jei $y \in \sigma(C)$, tai egzistuoja toks $c \in C$, kad $y = \sigma(c)$, ir $c$ perbėga $C$ tada ir tik tada, kai $y = \sigma(c)$ perbėga $\sigma(C)$. (2.9) ir (2.10) yra ekvivalentūs, nes skaliarinė sandauga nepasikeičia, jei abiejų vektorių koordinates perstatome, naudodami tą pačią perstatą. Šiuo atveju naudojome perstatą $\sigma^{-1}$ ir pasinaudojome tuo, kad $\sigma^{-1}(\sigma(c)) = c$. □

### 2.4.3 Kontrolinės matricos apibrėžimas ir savybės

**2.4.9 apibrėžimas.** *Tiesinio kodo* kontroline matrica *vadiname jo dualaus kodo generuojančią matricą.*

Kaip ir generuojančią matricą, taip ir kontrolinę matricą kodas gali turėti ne vieną.

Tiesinio kodo kontrolinė matrica, kaip ir generuojanti matrica, vienareikšmiškai apibrėžia kodą. Kartais yra patogiau nurodyti kodą pateikiant kontrolinę, o ne generuojančią matricą, nes kontrolinė matrica leidžia nesunkiai patikrinti, ar duotas vektorius priklauso kodui (todėl ir vadinasi „kontroline“). Iš tikro, galioja toks teiginys.

Čia ir toliau $y^T$ žymi transponuotą vektorių ar matricą $y$.

**2.4.10 teiginys.** *Tarkime, $H$ yra tiesinio kodo $C[n,k]$ virš $\mathbb{F}_q$ kontrolinė matrica, $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$. Vektorius $x \in C$ tada ir tik tada, kai $Hx^T = 0$.*

*Įrodymas.* Atkreipkime dėmesį, kad jei $H_1, \ldots, H_{n-k}$ pažymėtume matricos $H$ eilutes, tai

$$Hx^T = (H_1 \cdot x, \ldots, H_{n-k} \cdot x)^T, \tag{2.11}$$

čia $H_i \cdot x$ yra vektorių skaliarinė sandauga.

Visų pirma parodykime, kad jei $Hx^T = 0$, tai $x \in C$. Pagal (2.11) lygybę, $Hx^T = 0$ reiškia, kad vektorius $x$ yra ortogonalus kiekvienai matricos $H$ eilutei. Bet tokiu atveju vektorius $x$ yra ortogonalus kiekvienam kodo $C^{\perp}$ vektoriui $y$, nes $y$ gali būti išreikštas kodo $C^{\perp}$ bazės vektorių (t. y. matricos $H$ eilučių) tiesine kombinacija $y = a_1 H_1 + \cdots + a_{n-k} H_{n-k}$, ir tada

$$y \cdot x = (a_1 H_1 + \cdots + a_{n-k} H_{n-k}) \cdot x = a_1 (H_1 \cdot x) + \cdots + a_{n-k} (H_{n-k} \cdot x) = 0 + \cdots + 0 = 0.$$



Todėl pagal dualaus kodo apibrėžimą $x \in (C^{\perp})^{\perp} = C$.

Dabar įrodysime į kitą pusę. Kadangi $C = (C^{\perp})^{\perp}$, tai $x \in C$ reiškia, kad $x \in (C^{\perp})^{\perp}$. Pagal dualaus kodo apibrėžimą, vektorius $x$ yra ortogonalus kiekvienam kodo $C^{\perp}$ vektoriui, o tuo pačiu ir matricos $H$ eilutėms $H_1, \ldots, H_{n-k}$, todėl $Hx^T = 0$. □

Nesunku pastebėti, kad, kadangi $H$ yra $(n-k) \times n$ matrica, $x^T$ yra $n$ ilgio vektorius-stulpelis, tai $Hx^T$ yra $n-k$ ilgio vektorius-stulpelis.

**2.4.11 pavyzdys.** Tarkime, dvejetainio tiesinio kodo $C$ kontrolinė matrica yra

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

Patikrinkime, pavyzdžiui, ar vektorius $x = (010)$ priklauso kodui $C$:

$$Hx^T = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Todėl $x \notin C$. □

### 2.4.4 Kontrolinės matricos radimas

Dabar parodysime, kaip nesunkiai galima rasti kodo kontrolinę matricą. Žymėkime $I_k$ vienetinę $k \times k$ matricą. Jei $B$ ir $B'$ yra tiek pat eilučių turinčios matricos, tai $(B|B')$ bus matrica, gauta sujungus abi matricas į vieną (tiesiog prie matricos $B$ stulpelių prijungiame matricos $B'$ stulpelius). Jei $B$ ir $B'$ yra atitinkamai $k \times n_1$ ir $k \times n_2$ matricos, tai $(B|B')$ bus $k \times (n_1 + n_2)$ matrica.

**2.4.12 teiginys.** *Jei $G = (I_k|A)$ yra kodo $C$ generuojanti matrica, tai $H = (-A^T|I_{n-k})$ yra kodo $C$ kontrolinė matrica.*

Be įrodymo.

Taigi, suvedę kodo generuojančią matricą į standartinį pavidalą, galime rasti kontrolinę matricą.

**2.4.13 pavyzdys.** Rasime 2.3.2 pavyzdžio kodo $C$ kontrolinę matricą $H$. Tam turime suvesti generuojančią matricą $G$ į standartinį pavidalą ir pasinaudoti 2.4.12 teiginiu. Matrica $G$ jau buvo suvesta į standartinį pavidalą 2.3.7 pavyzdyje:

$$G = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \sim \left(\begin{array}{cc|cc} 1 & 0 & 1 & 2 \\ 0 & 1 & 1 & 1 \end{array}\right).$$

Todėl kontrolinė matrica

$$H = \left(\begin{array}{cc|cc} -1 & -1 & 1 & 0 \\ -2 & -1 & 0 & 1 \end{array}\right) = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix}.$$ □

2.4.12 teiginys leidžia rasti kontrolinę matricą, jei kodas turi standartinio pavidalo generuojančią matricą. Jei kodas neturi standartinio pavidalo generuojančios matricos, pasinaudojame tuo, kad kiekvienas kodas ekvivalentus kodui, turinčiam standartinio pavidalo generuojančią matricą, ir 2.4.8 teiginiu.

**2.4.14 pavyzdys.** Matėme, kad 2.3.12 pavyzdžio pirmos dalies kodas $C$ neturi standartinio pavidalo generuojančios matricos. Raskime jo kontrolinę matricą.

Prisiminkime, kad bandydami suvesti kodo $C$ generuojančią matricą $G$ į standartinį pavidalą, gavome vienetinės matricos stulpelius 1, 2 ir 4 pozicijose:

$$G = \begin{pmatrix} 4 & 3 & 1 & 4 & 3 \\ 3 & 1 & 2 & 0 & 4 \\ 4 & 1 & 4 & 2 & 4 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 2 & 0 & 2 \\ 0 & 1 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{pmatrix}.$$

Pritaikę kodui $C$ perstatą

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix},$$

gauname kodui $C$ ekvivalentų kodą $C' = \sigma(C)$, kurio generuojanti matrica

$$G' = \left(\begin{array}{ccc|cc} 1 & 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 1 & 3 \\ 0 & 0 & 1 & 0 & 4 \end{array}\right)$$

yra standartinio pavidalo. Pasinaudoję 2.4.12 teiginiu, galime rasti kodo $C'$ kontrolinę matricą:

$$H' = \left(\begin{array}{ccc|cc} -2 & -1 & 0 & 1 & 0 \\ -2 & -3 & -4 & 0 & 1 \end{array}\right) = \begin{pmatrix} 3 & 4 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 & 1 \end{pmatrix}.$$

Kadangi $C' = \sigma(C)$, tai pagal 2.4.8 teiginį $C'^{\perp} = \sigma(C^{\perp})$, t. y. $C^{\perp} = \sigma^{-1}(C'^{\perp})$. Taigi, pritaikę kodui $C'^{\perp}$ atvirkštinę perstatą

$$\sigma^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 4 & 3 & 5 \end{pmatrix},$$

gausime mūsų ieškomą dualų kodą $C^{\perp}$. Tai reiškia, kad matrica

$$H = \begin{pmatrix} 3 & 4 & 1 & 0 & 0 \\ 3 & 2 & 0 & 1 & 1 \end{pmatrix},$$

gauta sukeitus matricos $H'$ stulpelius pagal perstatą $\sigma^{-1}$, bus kodo $C^{\perp}$ generuojanti matrica, t. y. kodo $C$ kontrolinė matrica. □

**2.4.15 užduotis.** Rasti 1.2 poskyryje pateiktų paprastų pavyzdžių kontrolines matricas (pasinaudoti 2.3.14 pavyzdžiu).

**2.4.16 pastaba.** Tarkime, turime tiesinio kodo $C$ kontrolinę matricą $H$ ir norime rasti generuojančią matricą $G$. Matrica $H$ yra kodo $C^{\perp}$ generuojanti matrica. Pagal 2.4.6 teiginį $(C^{\perp})^{\perp} = C$, todėl matrica $G$ yra kodo $C^{\perp}$ kontrolinė matrica. Taigi, matricą $G$ galime rasti pasinaudoję 2.4.12 teiginiu: suvedame matricą $H$ į standartinį pavidalą ir randame $G$.

**2.4.17 pavyzdys.** Jei tiesinio kodo virš $\mathbb{F}_3$ kontrolinė matrica yra

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

tai generuojančią matricą $G$ randame lygiai taip pat, kaip 2.4.13 pavyzdyje. Gauname, kad

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



Matėme, kad rasti tiesinio kodo kontrolinę matricą nėra sunku. Taip pat, kaip rodo kitas teiginys, nėra sunku ir nustatyti, ar duota matrica yra duoto kodo kontrolinė matrica, ar ne.

**2.4.18 teiginys.** *Tegu $C$ yra tiesinis $[n, k]$ kodas virš $\mathbb{F}_q$, generuotas matricos $G$, o $H$ yra matrica virš $\mathbb{F}_q$. Matrica $H$ yra kodo $C$ kontrolinė matrica tada ir tik tada, kai $H$ yra $(n-k) \times n$ matrica, jos rangas yra $n-k$, ir $GH^T = 0$ (čia $0$ yra $k \times (n-k)$ matrica, sudaryta vien iš nulių).*

**2.4.19 užduotis.** Įrodyti teiginį.

*Sprendimas.* Pastebėsime, kad sandaugos $U = GH^T$ $i$-tojoje eilutėje bei $j$-ajame stulpelyje stovintis elementas $u_{ij}$ yra matricos $G$ $i$-tosios eilutės ir matricos $H$ $j$-osios eilutės skaliarinė sandauga.

" $\Longrightarrow$ " Išplaukia iš 2.4.4 teoremos (dualaus kodo dimensija, kartu ir kontrolinės matricos eilučių skaičius, yra $n-k$), generuojančios matricos apibrėžimo (kontrolinės matricos visos eilutės yra tiesiškai nepriklausomi vektoriai, todėl jos rangas yra $n-k$), ir iš to, kad visi kodo $C$ žodžiai (įskaitant ir jo generuojančios matricos $G$ eilutes) yra ortogonalūs visiems kodo $C^{\perp}$ žodžiams (tuo pačiu ir jo generuojančios matricos $H$ eilutėms).

" $\Longleftarrow$ " Kadangi $(n-k) \times n$ matricos $H$ rangas yra $n-k$, tai jos eilutės yra tiesiškai nepriklausomos. Pažymėkime $D$ matricos $H$ generuotą kodą. Kaip ir 2.4.10 teiginio įrodyme, gauname, kad, kadangi $GH^T = 0$, tai visi kodo $C$ žodžiai yra ortogonalūs visiems kodo $D$ žodžiams, todėl $D \subseteq C^{\perp}$. Be to, $\dim D = n - k = \dim C^{\perp}$. Todėl $D = C^{\perp}$ ir $H$ yra kodo $C$ kontrolinė matrica. □

**2.4.20 pavyzdys.** Tegu $C[3, 1]$ yra dvejetainis tiesinis kodas, generuotas matricos $G = (111)$. Patikrinkime, ar matrica

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

yra kodo $C$ kontrolinė matrica. Iš tikro, nesunku įsitikinti, kad matrica $H$ tenkina teiginio reikalavimus: tai $2 \times 3$ dvejetainė matrica, jos rangas yra 2, ir $GH^T = 0$. □

## 2.4.5 Kontrolinė matrica ir minimalus atstumas

Kodo kontrolinė matrica gali praversti ir nustatant kodo minimalų atstumą.

**2.4.21 teorema.** *Tegu $H$ yra tiesinio kodo $C$ kontrolinė matrica. Kodo $C$ minimalus atstumas yra lygus $d$ tada ir tik tada, kai egzistuoja $d$ tiesiškai priklausomų matricos $H$ stulpelių, o bet kuri $d-1$ šios matricos stulpelių sistema yra tiesiškai nepriklausoma.*

Be įrodymo.

Kitaip tariant, kodo $C$ minimalus atstumas $d$ yra toks mažiausias skaičius, kad egzistuoja $d$ tiesiškai priklausomų kodo $C$ kontrolinės matricos $H$ stulpelių.

**2.4.22 pavyzdys.** Tegu $C[3, 1]$ yra dvejetainis tiesinis kodas, kurio kontrolinė matrica yra

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

Norėdami rasti kodo $C$ minimalų atstumą, pradėdami nuo $d = 1$ ieškokime tokio $d$, kad egzistuotų $d$ tiesiškai priklausomų matricos $H$ stulpelių.

Tegu $d = 1$. Ar egzistuoja $d$ tiesiškai priklausomų matricos $H$ stulpelių? Kaip matėme 2.1.2 poskyryje, aibė iš vieno vektoriaus yra tiesiškai priklausoma tik tada, kai tas vektorius yra nulinis. Matrica $H$ nulinių stulpelių neturi, tai visos aibės iš vieno stulpelio yra tiesiškai nepriklausomos. Todėl kodo $C$ minimalus atstumas nėra 1.

Tegu $d = 2$. Ar egzistuoja $d$ tiesiškai priklausomų matricos $H$ stulpelių? Kaip matėme 2.1.2 poskyryje, dviejų dvejetainių vektorių rinkinys yra tiesiškai priklausomas tada ir tik tada, kai tie vektoriai yra lygūs. Bet matrica $H$ lygių stulpelių neturi, tai visos aibės iš dviejų stulpelių yra tiesiškai nepriklausomos. Todėl kodo $C$ minimalus atstumas nėra 2.

Lieka patikrinti $d = 3$. Ar egzistuoja $d$ tiesiškai priklausomų matricos $H$ stulpelių? Taip, nes matome, kad trečias stulpelis yra pirmų dviejų suma, o tai ir reiškia, kad šie trys stulpeliai yra tiesiškai priklausomi. Todėl kodo $C$ minimalus atstumas yra 3.

Ir iš tikro, nesunku įsitikinti, kad kodo $C$ generuojanti matrica yra $G = (111)$, todėl $C = \{000, 111\}$ — aišku, kad mažiausias atstumas tarp skirtingų kodo $C$ žodžių yra 3. □

### 2.4.6 Savidualūs kodai

**2.4.23 apibrėžimas.** *Jei $C \subseteq C^\perp$, kodas $C$ vadinamas* silpnai savidualiu. *Jei $C = C^\perp$, kodas $C$ vadinamas* (griežtai) savidualiu.

**2.4.24 pavyzdys.** Dvejetainis pakartojimo kodas $R_n$ yra silpnai savidualus tada ir tik tada, kai $n$ yra lyginis. Iš tikrųjų, $R_n = \{0\cdots0, 1\cdots1\}$, kur žodžių ilgis yra $n$. Kada $R_n$ yra silpnai savidualus, t. y. kada $R_n \subseteq R_n^\perp$? Aišku, kad $0\cdots0$ visada priklauso $R_n^\perp$, nes $R_n^\perp$ yra tiesinis kodas. Lieka panagrinėti, kada $1\cdots1 \in R_n^\perp$. Pagal dualaus kodo apibrėžimą, $1\cdots1 \in R_n^\perp$ reiškia, kad $1\cdots1$ yra ortogonalus visiems kodo $R_n = \{0\cdots0, 1\cdots1\}$ žodžiams. Žodžiui $0\cdots0$ jis bus ortogonalus visada. Belieka nustatyti, kada jis bus ortogonalus žodžiui $1\cdots1$, t. y. sau pačiam. Aišku, kad $1\cdots1$ bus ortogonalus sau pačiam tada ir tik tada, kai $n$ yra lyginis.

Kai $n = 2$, tai dvejetainis pakartojimo kodas $R_2 = \{00, 11\}$ yra savidualus. Patikrinkite patys.□

**2.4.25 teiginys.** *Tegu $C[n, k]$ yra tiesinis kodas, generuotas matricos $G$.*

1. *Kodas $C$ yra silpnai savidualus tada ir tik tada, kai $GG^T = 0$.*
2. *Kodas $C$ yra savidualus tada ir tik tada, kai $k = n/2$ ir $GG^T = 0$.*
3. *Kodas $C$ yra savidualus tada ir tik tada, kai $G$ yra kodo $C$ kontrolinė matrica.*

**2.4.26 užduotis.** Įrodyti teiginį.

*Sprendimas.* 1. Tarkime, tiesinis kodas $C$ yra silpnai savidualus. Tada pagal apibrėžimą $C \subseteq C^\perp$. Taigi, visi kodo $C$ žodžiai yra ortogonalūs visiems kodo $C$ žodžiams (įskaitant ir patiems sau). Tuo pačiu ir visos generuojančios matricos $G$ eilutės yra ortogonalios visoms $G$ eilutėms, todėl $GG^T = 0$.

Tarkime, kad $GG^T = 0$, t. y. visos generuojančios matricos $G$ eilutės yra ortogonalios visoms $G$ eilutėms. Kaip ir 2.4.10 teiginio įrodyme, gauname, kad visi kodo $C$ žodžiai yra ortogonalūs visiems kodo $C$ žodžiams, todėl $C \subseteq C^\perp$ ir $C$ yra silpnai savidualus.

2. Tarkime, tiesinis $[n, k]$ kodas $C$ yra savidualus. Tada pagal apibrėžimą $C = C^\perp$, todėl $\dim C = \dim C^\perp$. Bet $\dim C^\perp = n - \dim C$, todėl $\dim C = n - \dim C$, t. y. $k = \dim C = n/2$. Be to, $C$ yra silpnai savidualus, todėl pagal 1 dalį $GG^T = 0$.

Tarkime, turime tokį tiesinį $[n, k]$ kodą $C$, generuotą matricos $G$, kurio dimensija $k = n/2$ ir $GG^T = 0$. Pagal 1 dalį $C$ yra silpnai savidualus, t. y. $C \subseteq C^\perp$. Be to, $\dim C^\perp = n - k = n - n/2 = n/2 = \dim C$, todėl $C = C^\perp$ ir kodas $C$ yra savidualus.



3. Tarkime, tiesinis $[n, k]$ kodas $C$, generuotas matricos $G$, yra savidualus. Pagal 2 dalį $k = n/2$ ir $GG^T = 0$. Todėl pagal 2.4.18 teiginį matrica $G$ yra kodo $C$ kontrolinė matrica.

Tarkime, turime tiesinį kodą $C$, kurio generuojanti matrica $G$ yra ir kontrolinė kodo $C$ matrica. Tai reiškia, kad $k = n - k$, todėl $k = n/2$. Be to, pagal 2.4.18 teiginį $GG^T = 0$, todėl pagal 2 dalį kodas $C$ yra savidualus. □

**2.4.27 pavyzdžiai.** 1. Tegu $C[4, 1]$ yra dvejetainis tiesinis kodas, generuotas matricos $G = (1111)$. Pagal teiginio 1 dalį jis yra silpnai savidualus, nes $GG^T = 0$.

2. Tegu $C[4, 2]$ yra dvejetainis tiesinis kodas, generuotas matricos

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

Pagal teiginio 2 dalį jis yra savidualus, nes $k = 2 = 4/2 = n/2$ ir $GG^T = 0$. □

## 2.5 Tiesinių kodų dekodavimas

Tegu $C[n, k, d]$ yra tiesinis kodas virš $\mathbb{F}_q$ (taigi, $C \subset \mathbb{F}_q^n$), generuotas matricos $G$. Pranešimo $m \in \mathbb{F}_q^k$ kodavimas-dekodavimas gali būti parodytas tokia schema:

$$m \in \mathbb{F}_q^k \xrightarrow{\text{kodavimas}} c = mG \in C \subset \mathbb{F}_q^n \xrightarrow{\text{kanalas}} y = c + e \in \mathbb{F}_q^n \xrightarrow{\text{dekodavimas}} c' \in C \xrightarrow{c'=m'G} m' \in \mathbb{F}_q^k$$

Kanale prie kodo žodžio $c$ pridedamas klaidų vektorius $e \in \mathbb{F}_q^n$, ir iš kanalo gauname vektorių $y = c + e \in \mathbb{F}_q^n$. Dekodavimo metu paprastai visų pirma randame klaidų vektorių $e' \in \mathbb{F}_q^n$ (tikimės, kad $e' = e$), atėmę jį iš $y$ gauname kodo žodį $c' = y - e' \in C$, o tada randame pranešimą $m' \in \mathbb{F}_q^k$, kuris užkoduojamas kodo žodžiu $c'$. Pranešimas $m'$ randamas pasinaudojus tuo, kad $c' = m'G$. Tai tiesinių lygčių sistema, kuri, jei $c' \in C$, turi lygiai vieną sprendinį. Jei teisingai nustatėme klaidų vektorių (t. y. jei $e' = e$), tai rasime ir teisingą pranešimą (t. y. gausime $m' = m$), o jei ne, tai ne.

Šiame skyriuje pateikiame vieną dekodavimo procedūrą, kuri tinka bet kuriam tiesiniam kodui. Tai dekodavimas, naudojantis standartinę lentelę. Ši procedūra leidžia rasti kodo žodį $c' \in C$, esantį arčiausiai iš kanalo išėjusio vektoriaus $y$, t. y. realizuoja minimalaus atstumo dekodavimo taisyklę. Tokiu būdu ši procedūra leidžia ištaisyti visas klaidas, jei jų skaičius neviršija $t = [(d - 1)/2]$ (o kartais leidžia ištaisyti ir kai viršija).

### 2.5.1 Klasės

**2.5.1 apibrėžimas.** *Tegu $C[n, k, d]$ yra tiesinis kodas virš $\mathbb{F}_q$. Tegu $a \in \mathbb{F}_q^n$.* Klase *vadinsime aibę*

$$a + C = \{a + x : x \in C\}.$$

**2.5.2 pavyzdys.** Tegu $C = \{000, 111\}$ yra dvejetainis tiesinis kodas (tai pakartojimo kodas $R_3$). Raskime visas klases. 2.4 lentelėje surašyti visi vektoriai $a \in \mathbb{F}_2^3$ ir atitinkamos klasės $a + C$. Matome, kad $000 + C = 111 + C = \{000, 111\}$ ir t.t. Taigi, iš viso yra 4 klasės $\{000, 111\}$, $\{001, 110\}$, $\{010, 101\}$ ir $\{100, 011\}$. Matome, kad viena iš klasių yra pats kodas. □

| $a$ | $a+C$ |
|---|---|
| 000 | $\{000, 111\}$ |
| 001 | $\{001, 110\}$ |
| 010 | $\{010, 101\}$ |
| 100 | $\{100, 011\}$ |
| 011 | $\{011, 100\}$ |
| 101 | $\{101, 010\}$ |
| 110 | $\{110, 001\}$ |
| 111 | $\{111, 000\}$ |

2.4 lentelė: Klasių pavyzdys

**2.5.3 teiginys.** *1. Kiekvienas vektorius $b \in \mathbb{F}_q^n$ priklauso kuriai nors klasei. Tiksliau, $b \in b+C$.*

*2. Vektoriai $a, b \in \mathbb{F}_q^n$ priklauso tai pačiai klasei tada ir tik tada, kai $a-b \in C$.*

*3. Kiekvienai klasei priklauso $q^k$ vektorių.*

*4. Klasės arba nesikerta, arba sutampa.*

*5. Tarkime, kad kodo $C$ minimalus atstumas yra $d$. Tada kiekvienoje klasėje egzistuoja ne daugiau kaip vienas žodis, kurio svoris yra mažesnis už $d/2$.*

**2.5.4 užduotis.** Įrodyti teiginį.

*Sprendimas.* 1. Kadangi $C$ yra tiesinis kodas, tai $0 \in C$, todėl $b = b+0 \in b+C$.

2. "⇒": Tarkime, $a, b \in c+C$ . Tada egzistuoja tokie $c_1, c_2 \in C$, kad $a = c+c_1$ ir $b = c+c_2$. Taigi, $a-b = c_1-c_2$. Bet $C$ yra tiesinis kodas, todėl $c_1 - c_2 \in C$.

   "⇐": Tarkime, $a - b \in C$. Tada $a \in b + C$. Bet $b \in b + C$, todėl $a$ ir $b$ priklauso tai pačiai klasei.

3. Žinome, kad $|C| = q^k$ (teorema iš 2.2 poskyrio). Aišku, kad klasės $a + C$ vektorių skaičius irgi neviršija $q^k$. Be to, jis negali būti ir mažesnis, nes jei $c_1, c_2 \in C, c_1 \neq c_2$, tai $a + c_1 \neq a + c_2$.

4. Imkime dvi klases: $a+C$ ir $b+C$. Jei jos nesikerta — teiginys įrodytas. Tarkime, kad jos kertasi, t. y. kad jų sankirta nėra tuščia. Reikia parodyti, kad jos sutampa. Tegu $v \in a+C$ ir $v \in b+C$. Tada $v = a+c_1$, $c_1 \in C$, ir $v = b + c_2$, $c_2 \in C$, todėl $a + c_1 = b + c_2$, t. y. $a = b + c_2 - c_1 \in b + C$ (vėlgi todėl, kad kodas $C$ tiesinis, gauname, kad $c_2 - c_1 \in C$). Taigi, $a + C \subseteq b + C$. Lygiai taip pat išreiškę $b$ per $a$ gausime $b + C \subseteq a + C$. Vadinasi, $a + C = b + C$.

5. Tarkime $x$ ir $y$ priklauso tai pačiai klasei $x+C$, ir $w(x) < \frac{d}{2}$, $w(y) < \frac{d}{2}$. Pagal trikampio nelygybę, $w(x-y) = d(x,y) \leqslant d(x,0) + d(0,y) = w(x) + w(y) < \frac{d}{2} + \frac{d}{2} = d$. Bet šio teiginio antras punktas rodo, kad $x - y \in C$. Taigi, radome kodo $C$ žodį $x - y$, kurio svoris mažesnis už kodo minimalų atstumą $d$. Taip gali būti tik tuo atveju, kai $x - y = 0$. Taigi, $x = y$. □

Teiginys tvirtina, kad erdvę $\mathbb{F}_q^n$ galima padalinti į $r$ tarpusavyje nesikertančių klasių:

$$\mathbb{F}_q^n = (a_0 + C) \cup (a_1 + C) \cup \cdots \cup (a_{r-1} + C),$$

kur $r = \left|\mathbb{F}_q^n\right| / |C| = q^n/q^k = q^{n-k}$. Taip pat laikysime, kad $a_0 = 0$, todėl pirmoji klasė $a_0 + C = 0 + C = C$.



## 2.5.2 Dekodavimas

**2.5.5 teiginys.** *Galimų klaidų vektorių aibė sutampa su klase, kurioje yra iš kanalo gautas vektorius.*

*Įrodymas.* Tarkime, pranešimą $m \in \mathbb{F}_q^k$ užkoduojame kodu $C$, generuotu matricos $G$, gauname $c = mG \in C$. Iš kanalo gauname vektorių $y = c+e \in \mathbb{F}_q^n$. Tada klaidų vektorius $e = y-c \in y+C$, nes $-c \in C$. Taigi, klaidų vektorius $e$ priklauso tai pačiai klasei, kaip ir iš kanalo gautas vektorius $y$. Iš kitos pusės, bet kuris šios klasės vektorius galėtų būti klaidų vektoriumi. Iš tikrųjų, tarkime, $z \in y + C$. Galėjo būti, kad į kanalą buvo pasiųstas kodo žodis $y - z \in C$ (pagal 2.5.3 teiginio 2 dalį), kanale prie jo buvo pridėtas klaidų vektorius $z$. Tokiu atveju iš kanalo iš tiesų gauname $y$.□

Tarkime, kad dekodavimui naudojame minimalaus atstumo dekodavimo taisyklę, t. y. dekoduojame tuo kodo žodžiu $x \in C$, kuris yra arčiausiai iš kanalo gauto vektoriaus $y \in \mathbb{F}_q^n$, t. y. kuris tenkina sąlygą $d(x, y) = \min_{z \in C} d(z, y)$. Taigi, nusprendžiame, kad toks $x \in C$ yra tas vektorius, kuris buvo išsiųstas į kanalą, ir klaidų vektoriumi laikome $e = y - x$ (nes $y = x + e$).

Tą patį gausime ir darydami kitaip: ieškokime tokio klaidų vektoriaus $e$, kurio svoris būtų mažiausias. Tokiu atveju skirtumo $e = y-x$ svoris $w(y-x)$, o tuo pačiu ir atstumas tarp $x$ ir $y$ (nes $w(y - x) = d(x, y)$) irgi bus mažiausias. Taigi, taip darydami irgi naudojame minimalaus atstumo dekodavimo taisyklę. Prisiminę, kad klaidų vektorius priklauso vektoriaus $y$ klasei, gauname tokią dekodavimo procedūrą:

> *Klasėje, kuriai priklauso iš kanalo gautas vektorius $y$, randame mažiausio svorio vektorių $e$ (laikome jį klaidų vektoriumi), ir dekoduojame kodo $C$ žodžiu $y - e$.*

**2.5.6 apibrėžimas.** *Mažiausio svorio klasės vektorius vadinamas klasės* lyderiu. *Jei klasėje yra keli mažiausio svorio vektoriai, tai klasės lyderiu vadinsime kurį nors vieną iš jų.*

## 2.5.3 Standartinė lentelė

Taigi, norėdami dekoduoti, turime rasti klasės lyderį. Paprasčiausia yra visų klasių lyderius susirasti iš anksto, o dekodavimo metu tik pažiūrėti, kuriai klasei priklauso iš kanalo išėjęs vektorius, ir pasinaudoti jos lyderiu. Tai galime atlikti, sudarę standartinę lentelę.

Tarkime, $C[n, k, d]$ yra tiesinis kodas virš $\mathbb{F}_q$, generuotas matricos $G$. Sudarysime tokią lentelę. Pirmoje eilutėje surašome visus galimus pranešimų erdvės $\mathbb{F}_q^k$ žodžius $m_0, m_1, \ldots, m_{N-1}$, čia $N = q^k$. Tegu $m_0 = 00\cdots0$ — nulinis vektorius. Į antrą eilutę surašome atitinkamus kodo $C$ žodžius, t. y. $c_0 = 0, c_1, \ldots, c_{N-1}$, kur $c_i = m_iG$, $i = 0, 1, \ldots, N-1$. Trečiąją ir kitas lentelės eilutes užpildome taip: pasirenkame tokį mažiausio svorio vektorių $a \in \mathbb{F}_q^n$, kurio dar nebuvo prieš tai užrašytose eilutėse, ir jį užrašome pirmojoje vietoje, o paskui likusius klasės $a+C$ žodžius $a+c_1, a+c_2, \ldots, a+c_{N-1}$. Taip darome, kol užrašome visus erdvės $\mathbb{F}_q^n$ žodžius. Gauname tokią lentelę:

| Pranešimai: | $0$ | $m_1$ | $m_2$ | $\cdots$ | $m_{N-1}$ |
|---|---|---|---|---|---|
| Kodas: | $0$ | $c_1$ | $c_2$ | $\cdots$ | $c_{N-1}$ |
| Klasės: | $a_1$ | $a_1+c_1$ | $a_1+c_2$ | $\cdots$ | $a_1+c_{N-1}$ |
| | $a_2$ | $a_2+c_1$ | $a_2+c_2$ | $\cdots$ | $a_2+c_{N-1}$ |
| | $\vdots$ | $\vdots$ | $\vdots$ | $\cdots$ | $\vdots$ |
| | $a_{r-1}$ | $a_{r-1}+c_1$ | $a_{r-1}+c_2$ | $\cdots$ | $a_{r-1}+c_{N-1}$ |

Čia $r$ yra klasių skaičius. Ši lentelė vadinama *standartine kodo $C$ lentele*. Jos sudarymo būdas garantuoja, kad kiekvienoje eilutėje išrašyti atitinkamos klasės $a_i + C$, $i = 0, 1, \ldots, r-1$, vektoriai, o pirmasis iš jų yra klasės lyderis.

Turėdami standartinę kodo lentelę, dekoduojame iš kanalo gautą vektorių $y \in \mathbb{F}_q^n$ taip:

1. Randame, kurioje standartinės lentelės eilutėje yra $y$ (jis tikrai kažkur bus, nes lentelėje yra visi erdvės $\mathbb{F}_q^n$ vektoriai).

2. Nusprendžiame, kad šios eilutės pradžioje stovintis klasės lyderis $a$ yra klaidų vektorius, ir dekoduojame vektorių $y$ žodžiu $y - a$, t. y. žodžiu, kuris yra vektoriaus $y$ stulpelio viršuje.

**2.5.7 pavyzdys.** Tarkime, kodas $C$ yra dvejetainis tiesinis $[5, 2, 3]$ kodas, generuotas matricos

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

Jo standartinė lentelė bus, pavyzdžiui, tokia:

| Pranešimai: | 00 | 10 | 01 | 11 |
|---|---|---|---|---|
| Kodas: | 00000 | 10110 | 01011 | 11101 |
| Klasės: | 10000 | 00110 | 11011 | 01101 |
| | 01000 | 11110 | 00011 | 10101 |
| | 00100 | 10010 | 01111 | 11001 |
| | 00010 | 10100 | 01001 | 11111 |
| | 00001 | 10111 | 01010 | 11100 |
| | 00101 | 10011 | 01110 | 11000 |
| | 01100 | 11010 | 00111 | 10001 |

Kodo $C$ standartinė lentelė gali būti ir kitokia. Tai priklauso nuo to, kokia tvarka išrikiuoti pranešimai. Tėra reikalaujama, kad pirmoje vietoje stovėtų nulinis pranešimas, o likę gali būti išrikiuoti bet kaip. Pakeitus pranešimų išrikiavimo tvarką, keistųsi vietomis standartinės lentelės stulpeliai.

Standartinės lentelės eilutės, kuriose surašyti klasių vektoriai, irgi gali keistis vietomis. Pavyzdžiui, sudarinėdami šią standartinę lentelę, į pirmos klasės pirmą poziciją rašome bet kurį mažiausio svorio vektorių, kurio dar nėra lentelėje. Nulinio svorio vektorius 00000 lentelėje jau yra (tai kodo žodis). Bet nėra nei vieno svorio 1 žodžio. Tai galime iš jų rinktis bet kurį. Pasirinkome 10000, išrašėme jo klasę. Vėl galime rinktis bet kurį iš likusių svorio 1 žodžių, ir t.t. Kai baigiasi svorio 1 žodžiai, renkamės iš dar neužrašytų svorio 2 žodžių. Ir t.t. Taip darome, kol užrašome visus erdvės $\mathbb{F}_2^5$ žodžius. Žinome, kad klasių yra $r = q^{n-k} = 2^{5-2} = 8$, tai lentelėje bus 8 eilutės su erdvės $\mathbb{F}_2^5$ žodžiais.

Žymėkime $y$ iš kanalo gautą vektorių. Tarkime, $y = 11001$. Randame jį lentelėje, jis yra paskutiniame stulpelyje. Jo klasės lyderį 00100 (esantį vektoriaus $y$ eilutės pirmoje vietoje) laikysime klaidų vektoriumi, ir dekoduojame žodžiu 11101.

Jei $y = 01011$, randame jį tarp kodo žodžių, padarome išvadą, kad klaidų nebuvo, ir dekoduojame tuo pačiu žodžiu 01011.

Tarkime, $y = 10011$. Randame jį priešpaskutinėje eilutėje. Klaidų vektoriumi laikome tos klasės lyderį 00101, ir dekoduojame kodo žodžiu 10110.

Pastebėkime, kad paskutinėse dviejose klasėse nebėra lyderio vienareikšmiškumo. Pavyzdžiui, priešpaskutinės klasės lyderiu galėjome rinktis vektorių 11000. Tokiu atveju vektorių $y = 10011$



būtume dekodavę kitu kodo žodžiu $10011 - 11000 = 01011$. Taip yra dėl to, kad klasėje, esančioje priešpaskutinėje lentelės eilutėje, mažiausias vektoriaus svoris yra 2. Tai reiškia, kad jei $y$ priklauso tai klasei, tai kanale įvyko bent dvi klaidos. O tiek ištaisyti kodas negali, nes tai $t = [(d-1)/2] = [(3-1)/2] = 1$ klaidą taisantis kodas. Ir tikrai, vektorius $y = 10011$ vienodu atstumu nutolęs nuo dviejų kodo žodžių — 10110 ir 01011, todėl dekoduoti galime bet kuriuo iš jų. Ir dekoduojame būtent taip, kad klaidos vektorius būtų standartinės lentelės sudarymo metu pasirinktas klasės lyderis. □

### 2.5.4 Sindromai ir sumažinta standartinė lentelė

Be abejo, standartinės lentelės metodas tinka dekoduoti naudojant tik labai mažus tiesinius kodus, nes atmintyje reikia saugoti visus erdvės $\mathbb{F}_q^n$ žodžius. Naudojamos atminties kiekį būtų galima sumažinti, jei turėtume galimybę nesunkiai nustatyti, kuriai klasei priklauso iš kanalo gautas vektorius. Tokiu atveju pakaktų pasinaudoti tik pirmuoju lentelės stulpeliu. Tai leidžia padaryti sindromas.

**2.5.8 apibrėžimas.** *Tegu $H$ yra tiesinio kodo $C$ kontrolinė matrica, $y \in \mathbb{F}_q^n$. Žodžio $y$* sindromu *vadiname vektorių $s(y) = Hy^T \in \mathbb{F}_q^{n-k}$.*

**2.5.9 pavyzdys.** Jei tiesinio kodo virš $\mathbb{F}_3$ kontrolinė matrica yra

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

tai vektoriaus $y = 2221$ sindromas

$$s(y) = Hy^T = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}.$$

□

**2.5.10 teiginys.** *Tegu $H$ yra tiesinio kodo $C$ kontrolinė matrica. Tegu $y \in \mathbb{F}_q^n$.*

1. *Vektorius $y \in C$ tada ir tik tada, kai $s(y) = 0$.*

2. *Du vektoriai priklauso tai pačiai klasei tada ir tik tada, kai jų sindromai lygūs.*

3. *Tegu $q = 2$ (t. y. $C$ yra dvejetainis kodas), o $y$ yra iš kanalo gautas galbūt iškraipytas vektorius. Tada $s(y)$ yra lygus sumai tų kontrolinės matricos $H$ stulpelių, kur įvyko klaidos.*

*Įrodymas.* 1. Išplaukia iš 2.4.10 teiginio ir sindromo apibrėžimo.

2. Jei vektoriai $y$ ir $z$ priklauso tai pačiai klasei, tai pagal 2.5.3 teiginio 2 dalį $y - z \in C$. Tada pagal 2.4.10 teiginį $H(y-z)^T = 0$, t. y. $Hy^T = Hz^T$, todėl $s(y) = s(z)$. Į kitą pusę įrodymas analogiškas.

3. Tarkime, $y = x + e$, kur $x \in C$ — į kanalą pasiųstas kodo žodis, $e = (e_1, \ldots, e_n) \in \mathbb{F}_q^n$ — klaidų vektorius. Kadangi $q = 2$, tai $e_i$ gali būti tik 0 ar 1. Be to, $x \in C$, todėl $Hx^T = 0$ (2.4.10 teiginys). Žymėkime $H_i$, $i = 1, \ldots, n$, matricos $H$ stulpelius. Tada

$$s(y) = Hy^T = H(x+e)^T = Hx^T + He^T = He^T = \sum_{i=1}^{n} e_i H_i = \sum_{\substack{1 \leqslant i \leqslant n \\ e_i = 1}} H_i.$$

Paskutinė suma yra sudaryta iš tų kontrolinės matricos stulpelių $H_i$, kuriems atitinkamas $e_i = 1$, t. y. kur įvyko klaidos. □

Paskutinio teiginio 2 dalis rodo, kad tarp sindromų aibės ir klasių aibės egzistuoja abipusiškai vienareikšmė atitiktis. Todėl kiekvienos standartinės lentelės eilutės gale galime prirašyti sindromą, atitinkantį toje eilutėje išrašytą klasę.

Gavę iš kanalo vektorių $y$, apskaičiuojame jo sindromą $s(y)$, kuris ir parodo, kurioje standartinės lentelės eilutėje $y$ yra. Taigi, norint dekoduoti, pakanka turėti tokią *sumažintą standartinę lentelę*:

| Klasių lyderiai | Sindromai |
|---|---|
| 0 | 0 |
| $a_1$ | $s_1$ |
| $a_2$ | $s_2$ |
| $\vdots$ | $\vdots$ |
| $a_{r-1}$ | $s_{r-1}$ |

Gavę iš kanalo vektorių $y$, apskaičiuojame jo sindromą $s(y)$, randame $s_i = s(y)$ sumažintoje standartinėje lentelėje, nusprendžiame, kad atitinkamas klasės lyderis $a_i$ yra klaidų vektorius ir dekoduojame vektorių $y$ kodo žodžiu $y - a_i$.

Sumažintą standartinę lentelę sudarome taip: imame visus erdvės $\mathbb{F}_q^n$ žodžius $y$, pradedant nuo mažiausio svorio vektorių, ir skaičiuojame jų sindromus $s(y)$. Jei gauname naują sindromą, tai $y$ ir $s(y)$ dedame į lentelę. Taip darome, kol gauname visas $q^{n-k}$ klases.

**2.5.11 pavyzdys.** Imkime kodą iš 2.4.17 pavyzdžio. Kaip matėme, tai kodas virš $\mathbb{F}_3$, kurio generuojanti ir kontrolinė matricos yra atitinkamai

$$G = \begin{pmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{pmatrix} \quad \text{ir} \quad H = \begin{pmatrix} 2 & 1 & 0 & 2 \\ 1 & 1 & 2 & 0 \end{pmatrix}.$$

Sudarysime sumažintą standartinę lentelę. Mūsų atveju klasių bus $3^{4-2} = 9$. Pradedame nuo svorio 0 vektoriaus $y = 0000$, kurio sindromas $s(y) = Hy^T = 00^T$, čia $00^T$ yra nulinis vektorius-stulpelis (transponuotas vektorius-eilutė). Toliau eina svorio 1 vektoriai $1000, \ldots, 0001, 2000, \ldots, 0002$, kurių sindromai yra atitinkamai $21^T, \ldots, 10^T$ (žr. lentelę). Visi jie skirtingi, dėl to visus juos dedame į lentelę. Gauname 9 eilutes, kiek ir turėjome gauti. Lentelę sudarėme. Kad jos sudarymo principas būtų aiškesnis, pažiūrėkime, ką būtume darę toliau, jei būtume radę dar ne visas eilutes. Toliau imtume svorio 2 vektorius. Pavyzdžiui, pradėtume nuo vektoriaus 1100, kurio sindromas yra $02^T$. Toks sindromas lentelėje jau yra, todėl vektorių 1100 atmetame (nes lentelėje jau turime jo klasės lyderį 0010). Taip bandytume visus kitus svorio 2 vektorius, paskui svorio 3 ir t. t., kol galų gale gautume tiek klasių, kiek turime gauti.



Gauname tokią sumažintą standartinę lentelę (paprastumo dėlei vektorius-stulpelius rašysime kaip atitinkamus vektorius-eilutes):

| Klasių lyderiai | Sindromai |
|---|---|
| 0000 | 00 |
| 1000 | 21 |
| 0100 | 11 |
| 0010 | 02 |
| 0001 | 20 |
| 2000 | 12 |
| 0200 | 22 |
| 0020 | 01 |
| 0002 | 10 |

Tarkime, iš kanalo gauname vektorių $y = 2221$. Apskaičiuojame jo sindromą $s(y) = 22^T$. Randame jį sumažintoje standartinėje lentelėje. Atitinkamą klasės lyderį 0200 laikome klaidų vektoriumi ir dekoduojame kodo žodžiu $2221 - 0200 = 2021$. □

### 2.5.5 Ribotas dekodavimas ir nepilna sumažinta standartinė lentelė

Visgi matome, kad tokią standartinę lentelę (sumažintą ar ne) sudaryti galime tik nedideliems $q$ ir $n$, nes turime peržiūrėti didelę dalį (ar visus) erdvės $\mathbb{F}_q^n$ žodžius. Tai būtų žymiai lengviau padaryti, jei apsiribotume nedideliu ištaisomų klaidų skaičiumi. Pavyzdžiui, norime ištaisyti visas klaidas, jei jų skaičius neviršija kažkokio iš anksto pasirinkto pakankamai nedidelio skaičiaus $K$. Jei viršija — ką gi, ištaisyti negalėsime, teks, pavyzdžiui, prašyti atsiųsti iš naujo. Tai galima būtų daryti taip.

2.5.3 teiginio 5 dalyje gavome, kad kiekvienoje klasėje egzistuoja ne daugiau kaip vienas žodis, kurio svoris yra mažesnis už $d/2$, kur $d$ yra kodo $C$ minimalus atstumas. Taigi jei klasėje egzistuoja nors vienas žodis, kurio svoris mažesnis už $d/2$, tai visų kitų klasės žodžių svoriai yra nemažesni už $d/2$, todėl šis žodis ir yra klasės lyderis.

Kadangi visi erdvės $\mathbb{F}_q^n$ žodžiai priklauso kuriai nors klasei, tai visi erdvės žodžiai, kurių svoris mažesnis už $d/2$, yra savo klasių lyderiai. Visiems tokiems žodžiams galima apskaičiuoti sindromus ir sudaryti *nepilną sumažintą standartinę lentelę* vien iš jų. Arba net iš dar mažiau žodžių: pasirenkame $K < d/2$, ir į nepilną sumažintą lentelę sudedame tik žodžius, kurių svoris nedidesnis už $K$, ir jų sindromus.

Dekoduojame tada taip: iš kanalo gavę vektorių $y$, apskaičiuojame jo sindromą $s(y)$, tada nepilnoje sumažintoje standartinėje lentelėje ieškome klasės lyderio $a_i$, kurio sindromas $s_i = s(y)$. Jei randame, tai vektorių $y$ dekoduojame žodžiu $y - a_i$, o jei nerandame, tai reiškia, kad kanale įvyko daugiau nei $K$ klaidų, ir mūsų naudojamas algoritmas neturi galimybių jų ištaisyti.

**2.5.12 pavyzdys.** Grįžkime prie kodo $C$, pateikto 2.5.7 pavyzdyje. Prisiminkime, kad tai dvejetainis tiesinis $[5, 2, 3]$ kodas, generuotas matricos

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

Nesunkiai galime rasti, kad kodas $C = \{00000, 10110, 01011, 11101\}$, todėl jo minimalus atstumas $d = 3$. Pasirinkime $K = 1$, $K < d/2$. Norint sudaryti sumažintą standartinę lentelę, dar reikia

žinoti kodo $C$ kontrolinę matricą $H$. Ją taip pat nesunkiai galime rasti:

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

Nepilnoje sumažintoje standartinėje lentelėje bus tik klasių lyderiai, kurių svoris nedidesnis už $K$, ir jų sindromai (paprastumo dėlei vektorius-stulpelius rašysime kaip atitinkamus vektorius-eilutes):

| Klasių lyderiai | Sindromai |
|---|---|
| 00000 | 000 |
| 10000 | 110 |
| 01000 | 011 |
| 00100 | 100 |
| 00010 | 010 |
| 00001 | 001 |

Tarkime, iš kanalo gauname vektorių $y = 11001$. Apskaičiuojame jo sindromą $s(y) = 100^T$. Randame jį nepilnoje sumažintoje standartinėje lentelėje. Atitinkamą klasės lyderį 00100 laikome klaidų vektoriumi ir dekoduojame kodo žodžiu $11001 - 00100 = 11101$.

Tarkime, iš kanalo gauname vektorių $y = 10011$. Apskaičiuojame jo sindromą $s(y) = 101^T$. Tokio sindromo nepilnoje sumažintoje standartinėje lentelėje nėra. Tai reiškia, kad kanale įvyko daugiau nei $K = 1$ klaida, ir mūsų naudojamas algoritmas neturi galimybių jų ištaisyti. Bet šiuo atveju tai nedidelis trūkumas, nes bet kokiu atveju kodas $C$ garantuotai gali ištaisyti tik $[(d-1)/2] = 1$ klaidą, t. y. nors pilna standartinė lentelė ir galėtų ištaisyti dvi klaidas, bet mes vistiek nebūtume garantuoti, kad ištaisė teisingai. □

Nepilna sumažinta standartinė lentelė žymiai lengviau sudaroma, todėl gali būti naudojama didesniems tiesiniams kodams, užtat jos klaidų taisymo galimybės yra ribotos.

Reziumuojant reikia pasakyti, kad yra parodyta, kad tiesinių kodų dekodavimo uždavinys yra NP-pilnas, t. y. vilties rasti polinominio laiko dekodavimo algoritmą, tinkantį visiems tiesiniams kodams, yra labai mažai. Todėl ieškoma tokių tiesinių kodų šeimos pošeimių, kuriems egzistuotų greitas polinominio laiko dekodavimo algoritmas. Kelias tokias šeimas (Hemingo kodus, Rydo-Miulerio kodus) mes ir panagrinėsime kituose skyriuose.



# 3 skyrius
# Kai kurios tiesinių kodų šeimos

## 3.1 Dvejetainiai Hemingo kodai

### 3.1.1 Apibrėžimas ir savybės

Hemingo vieną klaidą taisantys kodai yra svarbi klaidas taisančių tiesinių kodų šeima. Jais naudojantis, lengva ir užkoduoti, ir dekoduoti. Mes aptarsime tik dvejetainius Hemingo kodus.

**3.1.1 apibrėžimas.** *Tegu $r \geqslant 2$.* Dvejetainis Hemingo kodas $\mathbf{H}_2(r)$ *yra dvejetainis tiesinis ilgio $n = 2^r - 1$ kodas, kurio kontrolinės matricos stulpeliai yra visi galimi ilgio $r$ dvejetainiai skirtingi nenuliniai vektoriai.*

**3.1.2 pastaba.** Apibrėžimas nenustato, kuria tvarka tie stulpeliai turi būti išrikiuoti. Kad ir kaip jie būtų išrikiuoti, gautas kodas bus vadinamas Hemingo kodu. Taigi egzistuoja ištisa šeima dvejetainių ilgio $2^r - 1$ ekvivalenčių Hemingo kodų.

**3.1.3 pavyzdžiai.** 1. Tegu $r = 2$. Pagal apibrėžimą, Hemingo kodas $\mathbf{H}_2(2)$ yra dvejetainis tiesinis ilgio $n = 2^2 - 1 = 3$ kodas, kurio kontrolinės matricos $H$ stulpeliai yra visi galimi ilgio 2 dvejetainiai skirtingi nenuliniai vektoriai, t. y. vektoriai 10, 01 ir 11, todėl kontrolinė matrica yra, pavyzdžiui, tokia:

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

2. Tegu $r = 3$. Pagal apibrėžimą, Hemingo kodas $\mathbf{H}_2(3)$ yra dvejetainis tiesinis ilgio $n = 2^3 - 1 = 7$ kodas, kurio kontrolinės matricos $H$ stulpeliai yra visi galimi ilgio 3 dvejetainiai skirtingi nenuliniai vektoriai, todėl kontrolinė matrica yra, pavyzdžiui, tokia:

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

□

**3.1.4 teiginys.** *Dvejetainio Hemingo kodo minimalus atstumas $d = 3$.*

*Įrodymas.* Pagal 2.4.21 teoremą kodo minimalus atstumas lygus $d$, jei jo kontrolinės matricos $H$ bet kurie $d-1$ stulpeliai yra tiesiškai nepriklausomi ir egzistuoja $d$ tiesiškai priklausomų stulpelių.

Kadangi dvejetainio Hemingo kodo kontrolinės matricos visi stulpeliai yra nenuliniai ir skirtingi, tai bet kuri stulpelių pora yra tiesiškai nepriklausoma. Be to, į matricą įeina visi galimi ilgio $r$ nenuliniai dvejetainiai vektoriai, taigi bet kurių dviejų matricos $H$ stulpelių suma taip pat yra matricos $H$ stulpelis, nelygus nei vienam iš tų dviejų stulpelių, todėl šie trys stulpeliai yra tiesiškai priklausomi. Pagal 2.4.21 teoremą kodo minimalus atstumas $d = 3$. □

Taigi, dvejetainis Hemingo kodas $\mathbf{H}_2(r)$ yra tiesinis $[2^r - 1, 2^r - r - 1, 3]$ kodas.

**3.1.5 teiginys.** *Tegu $r \geqslant 2$. Bet kuris dvejetainis tiesinis $[2^r-1, 2^r-r-1, 3]$ kodas yra Hemingo kodas $\mathbf{H}_2(r)$.*

Įrodyti patiems.

**3.1.6 pavyzdys.** Kurso pradžioje, 1.2.6 poskyryje (14 psl.) apibrėžtas $[7, 4]$ Hemingo kodas - tai dvejetainis Hemingo kodas $\mathbf{H}_2(3)$. Iš tikrųjų, 2.3.14 pavyzdžio 6 dalyje (52 psl.) matėme, kad jo generuojanti matrica yra

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

Pagal 2.4.12 teiginį gauname jo kontrolinę matricą:

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

Matome, kad kontrolinės matricos $H$ stulpeliai iš tikrųjų yra visi galimi ilgio 3 dvejetainiai skirtingi nenuliniai vektoriai. □

### 3.1.2 Dekodavimas

Kadangi Hemingo kodo minimalus atstumas $d = 3$, tai $t = [(d-1)/2] = 1$, todėl Hemingo kodas taiso visas pavienes klaidas.

Pagal 2.5.10 teiginio 3 dalį, iš kanalo gauto vektoriaus $y$ sindromas lygus Hemingo kodo kontrolinės matricos $H$ stulpelių, atitinkančių klaidų pozicijas, sumai. Tarkime, įvyko lygiai viena klaida. Tada vektoriaus $y$ sindromas yra lygus tam kontrolinės matricos $H$ stulpeliui, kur įvyko klaida, todėl apskaičiavus sindromą užtenka rasti, kuris matricos $H$ stulpelis yra jam lygus ir ištaisyti klaidą tą stulpelį atitinkančioje vektoriaus pozicijoje. Formaliai dvejetainio Hemingo kodo $\mathbf{H}_2(r)$, $r \geqslant 2$, dekodavimo algoritmą galime užrašyti taip:

- Tarkime, iš kanalo gavome vektorių $y = (y_1, \ldots, y_n) \in \mathbb{F}_2^n$, kur $n = 2^r - 1$. Apskaičiuojame jo sindromą $s = s(y) \in \mathbb{F}_2^r$.
- Jei $s = 0$, tai $y \in C$ — algoritmas išveda $y$.
- Jei $s \neq 0$, tai $s$ yra kuris nors kontrolinės matricos $H$ stulpelis, tarkime, $j$-asis, $1 \leqslant j \leqslant n$ — algoritmas išveda $(y_1, \ldots, y_{j-1}, y_j + 1, y_{j+1}, \ldots, y_n)$.

Beje, šį algoritmą galima efektyviai realizuoti, jei kontrolinės matricos $H$ stulpeliai išrikiuoti tokia tvarka: $i$-tasis stulpelis — skaičiaus $i$ dvejetainė išraiška (jei reikia, papildyta nuliais), kur žemiausias narys yra kairėje, pavyzdžiui, vieneto dvejetainė išraiška yra $10\cdots0$. T. y.

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & \cdots \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{3.1}$$



Tada, apskaičiavus sindromą $s$, užtenka konvertuoti $s$ iš dvejetainės į dešimtainę sistemą ir gauname klaidos pozicijos numerį $j$. Tokiu atveju nereikia perbėgti visos matricos $H$, lyginant kiekvieną jos stulpelį su sindromu $s$.

**3.1.7 pavyzdys.** Naudodamiesi dvejetainiu Hemingo kodu $\mathbf{H}_2(3)$, kurio kontrolinė matrica $H$ yra (3.1) pavidalo, dekoduosime seką 0000010 1100110 0110100.

Kontrolinė matrica bus

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

Pažymėkime $y_1 = 0000010$. Sindromas $s(y_1) = 011^T$. Pavertę šią dvejetainę išraišką į dešimtainę gauname $j = 6$, t. y. klaida yra šeštoje pozicijoje. Todėl dekoduojame vektoriumi 0000000.

Toliau, tegu $y_2 = 1100110$. Tada $s(y_2) = 000^T$. Klaidų nėra, dekoduojame 1100110.

Na, ir $y_3 = 0110100$. Tada $s(y_3) = 001^T$, todėl $j = 4$ ir dekoduojame 0111100. □

## 3.2 Pirmos eilės Rydo-Miulerio kodai

### 3.2.1 Apibrėžimas ir savybės

**3.2.1 apibrėžimas.** *Tegu $m \geqslant 2$.*

1. *Funkcija $f : \mathbb{F}_2^m \to \mathbb{F}_2$ vadinama* logine funkcija *(arba* Būlio funkcija*).*
2. *Loginė funkcija*
$$f(x_1, x_2, \ldots, x_m) = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_m x_m + \mu,$$
*kur $\lambda_1, \lambda_2, \ldots, \lambda_m, \mu \in \mathbb{F}_2$, vadinama* afiniąja.

**3.2.2 pavyzdys.** Tegu $m = 3$. Funkcija $s(x_1, x_2, x_3) = x_1x_2 + x_1 + 1$ yra loginė funkcija. Jinai nėra afinioji, nes įeina kintamųjų sandauga $x_1x_2$. Funkcija $s'(x_1, x_2, x_3) = x_1 + x_3 + 1$ yra afinioji loginė funkcija. □

Kiekvienai loginei funkcijai galima sudaryti reikšmių lentelę, kurioje kintamųjų reikšmių rinkiniai išrikiuoti kaip 3.2.3 pavyzdyje.

**3.2.3 pavyzdys.** Paskutinio pavyzdžio loginės funkcijos $s(x_1, x_2, x_3) = x_1x_2 + x_1 + 1$ reikšmių lentelė yra tokia:

| $x_1$ | $x_2$ | $x_3$ | $s(x_1, x_2, x_3)$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

□

Loginės funkcijos įgyjamų reikšmių rinkinį iš reikšmių lentelės galima užrašyti ilgio $2^m$ vektoriumi. Pavyzdžiui, 3.2.3 pavyzdžio loginę funkciją $s(x_1, x_2, x_3) = x_1x_2 + x_1 + 1$ atitinka vektorius $\overrightarrow{s} = (s_0, \ldots, s_7) = (1, 1, 1, 1, 0, 0, 1, 1)$.

Formaliai tai galima užrašyti taip: loginę funkciją $f$ atitinka vektorius $\overrightarrow{f} = (f_0, \ldots, f_{2^m-1})$, kur $f_i = f(b_1, b_2, \ldots, b_m)$, o $b_1, \ldots, b_m$ yra skaičiaus $i$ dvejetainė išraiška (papildyta nuliais, jei reikia), t. y. $i = \sum_{j=0}^{m-1} b_{m-j}2^j$. Čia dvejetainės išraiškos žemiausias narys yra dešinėje, pavyzdžiui, vieneto dvejetainė išraiška yra $0 \cdots 01$. Taigi,

$$\begin{aligned} f_0 &= f(0, \ldots, 0, 0), \\ f_1 &= f(0, \ldots, 0, 1), \\ f_2 &= f(0, \ldots, 1, 0), \\ f_3 &= f(0, \ldots, 1, 1), \\ &\cdots \\ f_{2^m-1} &= f(1, \ldots, 1, 1). \end{aligned}$$

**3.2.4 teiginys.** *Jei $f$ ir $g$ yra loginės funkcijos, tai $\overrightarrow{f+g} = \overrightarrow{f} + \overrightarrow{g}$.*

**3.2.5 užduotis.** Įrodyti šį teiginį.

**3.2.6 apibrėžimas.** *Tegu $m \geqslant 2$.* Pirmos eilės dvejetainis Rydo-Miulerio (Reed-Muller) kodas, *žymimas $RM(1, m)$, yra vektorių, atitinkančių afiniąsias logines funkcijas $f : \mathbb{F}_2^m \to \mathbb{F}_2$, aibė.*

Taigi, kodo $RM(1, m)$ ilgis $n$ yra lygus vektorių ilgiui, t. y. $n = 2^m$. Afiniųjų loginių funkcijų yra $2^{m+1}$ (kiekvienam $\lambda_i$, $i = 1, \ldots, m$, parinkti turime dvi galimybes, kaip ir $\mu$). Be to, visas afiniąsias logines funkcijas atitinka skirtingi vektoriai (įrodykite patys). Taigi, kodo dydis (t. y. kodo žodžių skaičius) $|RM(1, m)|$ irgi yra $2^{m+1}$.

**3.2.7 pavyzdys.** Išrašykime visus kodo $RM(1, 3)$ žodžius. Tam reikės rasti visas galimas afiniąsias logines funkcijas $\mathbb{F}_2^3 \to \mathbb{F}_2$ ir sudaryti kiekvienos iš jų reikšmių lentelę. 3.1 lentelėje pateikiamos visos afiniosios loginės funkcijos ir jas atitinkantys vektoriai. Kairėje yra afiniosios loginės funkcijos $f$, kurių laisvasis narys $\mu$ lygus nuliui, o dešinėje jas atitinkančios $f + 1$, t. y. atitinkamos afiniosios loginės funkcijos su $\mu = 1$. Visi išvardinti $2^{3+1} = 16$ vektorių ir sudaro $RM(1, 3)$ kodą. □

Jei $\vec{f}, \vec{g} \in RM(1, m)$, tai $\vec{f} + \vec{g} = \overrightarrow{f+g}$. Bet $f + g$ irgi yra afinioji loginė funkcija, todėl $\overrightarrow{f+g} \in RM(1, m)$. Taigi, $RM(1, m)$ yra tiesinis kodas. Kadangi tai dvejetainis kodas, ir $|RM(1, m)| = 2^{m+1}$, tai kodo dimensija yra $m + 1$. Reziumuojant, $RM(1, m)$ yra tiesinis $[2^m, m + 1]$ kodas.

Raskime kodo $RM(1, m)$ generuojančią matricą.

**3.2.8 teiginys.** *Tegu $m \geqslant 2$. Matrica, kurios eilutės yra funkcijas $x_1, x_2, \ldots, x_m$ ir 1 atitinkantys vektoriai $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}, \vec{1}$, yra kodo $RM(1, m)$ generuojanti matrica.*

*Įrodymas.* Kad įrodytume, kad tikrai tokia matrica yra kodo $RM(1, m)$ generuojanti matrica, turime parodyti, kad vektoriai $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}, \vec{1}$ sudaro kodo $RM(1, m)$ bazę, t. y. reikia parodyti, kad jie priklauso kodui $RM(1, m)$ ir yra tiesiškai nepriklausomi. Kadangi jų skaičius yra lygus kodo $RM(1, m)$ dimensijai, tai ir gausime, kad jie sudaro bazę.



| Funkcija | Vektorius | Funkcija | Vektorius |
|---|---|---|---|
| $0$ | 00000000 | $1$ | 11111111 |
| $x_1$ | 00001111 | $x_1 + 1$ | 11110000 |
| $x_2$ | 00110011 | $x_2 + 1$ | 11001100 |
| $x_3$ | 01010101 | $x_3 + 1$ | 10101010 |
| $x_1 + x_2$ | 00111100 | $x_1 + x_2 + 1$ | 11000011 |
| $x_1 + x_3$ | 01011010 | $x_1 + x_3 + 1$ | 10100101 |
| $x_2 + x_3$ | 01100110 | $x_2 + x_3 + 1$ | 10011001 |
| $x_1 + x_2 + x_3$ | 01101001 | $x_1 + x_2 + x_3 + 1$ | 10010110 |

3.1 lentelė: $RM(1, 3)$ kodo žodžiai

Kadangi visi šie vektoriai atitinka afiniąsias funkcijas, jie priklauso $RM(1, m)$. Lieka parodyti, kad jie tiesiškai nepriklausomi. Imkime jų tiesinę kombinaciją $\sum_{i=1}^m \lambda_i\vec{x_i} + \mu \cdot \vec{1}$ su koeficientais $\lambda_1, \ldots, \lambda_m, \mu \in \mathbb{F}_2$ ir prilyginkime ją nuliui. Parodysime, kad tokiu atveju visi koeficientai yra lygūs nuliui. Kadangi $\vec{f}+\vec{g} = \overrightarrow{f+g}$ bet kurioms dviems loginėms funkcijoms $f$ ir $g$, tai $\sum_{i=1}^m \lambda_i\vec{x_i}+\mu\cdot\vec{1} = \overrightarrow{\sum_{i=1}^m \lambda_i x_i + \mu}$. Todėl $\overrightarrow{\sum_{i=1}^m \lambda_i x_i + \mu}$ irgi yra nulinis vektorius. Bet tėra viena afinioji funkcija, kurią atitinka nulinis vektorius — tai nulinė funkcija, todėl funkcija $\sum_{i=1}^m \lambda_i x_i + \mu$ ir yra nulinė funkcija, ir jos visi koeficientai $\lambda_1, \ldots, \lambda_m, \mu$ yra lygūs nuliui. Taigi, parodėme, kad jei vektorių $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}, \vec{1}$ tiesinė kombinacija yra lygi nuliui, tai visi tos tiesinės kombinacijos koeficientai yra lygūs nuliui. Tai ir reiškia, kad vektoriai yra tiesiškai nepriklausomi. □

**3.2.9 pavyzdys.** Kodo $RM(1, 3)$ generuojančios matricos eilutės yra vektoriai $\vec{x_1}, \vec{x_2}, \vec{x_3}, \vec{1}$. Loginės funkcijos 1 reikšmių lentelė yra tokia:

| $x_1$ | $x_2$ | $x_3$ | 1 |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

Pirmas stulpelis yra funkciją $x_1$ atitinkantis vektorius $\vec{x_1}$, antras — $\vec{x_2}$, trečias — $\vec{x_3}$, ir ketvirtas — $\vec{1}$. Todėl kodo $RM(1, 3)$ generuojanti matrica bus tiesiog transponuota ši reikšmių lentelė:

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

□

Žinome, kad koduodami tiesiniu kodu dauginame informacijos vektorių iš jo generuojančios matricos $G$, t. y. jei informacijos vektorius yra $u$, tai užkoduotas vektorius yra $uG$. Pasirodo, kad jei kodavimui Rydo-Miulerio kodu naudosime 3.2.8 teiginyje apibrėžtą generuojančią matricą, tai kodavimo procedūrą galėsime užrašyti ir kitaip. Atkreipkite dėmesį, kad informacijos vektoriaus $u$ ilgis yra lygus kodo dimensijai, kuri šiuo atveju yra $m + 1$.

**3.2.10 teiginys.** *Pažymėkime informacijos vektoriaus $u$ koordinates $\lambda_1, \ldots, \lambda_m$ ir $\mu$, t. y. $u = (\lambda_1, \ldots, \lambda_m, \mu)$. Tegu $G$ yra 3.2.8 teiginyje apibrėžta Rydo-Miulerio kodo generuojanti matrica. Tada $uG = \vec{f}$, kur $\vec{f}$ yra vektorius, atitinkantis loginę funkciją $f(x) = \sum_{i=1}^{m} \lambda_i x_i + \mu$.*

*Įrodymas.* Matricos $G$ eilutės yra $\vec{x_1}, \vec{x_2}, \ldots, \vec{x_m}, \vec{1}$, todėl

$$\begin{aligned} uG &= \lambda_1\vec{x_1} + \cdots + \lambda_m\vec{x_m} + \mu\vec{1} \\ &= \overrightarrow{\lambda_1 x_1 + \cdots + \lambda_m x_m + \mu} \\ &= \vec{f}. \end{aligned}$$

□

Taigi, užkoduotas vektorius yra $\vec{f}$.

**3.2.11 pavyzdys.** Tarkime, $m = 3$. Daugindami informacijos vektorių $u = (\lambda_1, \lambda_2, \lambda_3, \mu)$ iš 3.2.8 teiginyje pateiktos kodo $RM(1, m)$ generuojančios matricos $G$ (žr. 3.2.9 pavyzdį), gauname

$$\begin{aligned} &(\lambda_1, \lambda_2, \lambda_3, \mu) \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \\ &= (\mu, \lambda_3 + \mu, \lambda_2 + \mu, \lambda_2 + \lambda_3 + \mu, \lambda_1 + \mu, \lambda_1 + \lambda_3 + \mu, \lambda_1 + \lambda_2 + \mu, \lambda_1 + \lambda_2 + \lambda_3 + \mu) \\ &= (f(0,0,0), f(0,0,1), f(0,1,0), f(0,1,1), f(1,0,0), f(1,0,1), f(1,1,0), f(1,1,1)) \\ &= \vec{f}. \end{aligned}$$

□

## 3.2.2 Dekodavimas

Operatorių $\Delta_i$ apibrėžkime taip:

$$\Delta_i : f(x) \mapsto \Delta_i f(x) = f(x + a_i) + f(x),$$

kur $f(x)$ — loginė funkcija, $1 \leqslant i \leqslant m$, ir

$$a_i = (0, 0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{F}_2^m,$$

čia vienetas yra $i$-tojoje pozicijoje. Kitaip tariant,

$$\Delta_i f(x_1, \ldots, x_m) = f(x_1, \ldots, x_{i-1}, x_i + 1, x_{i+1}, \ldots, x_m) + f(x_1, \ldots, x_m).$$

**3.2.12 pavyzdys.** Tegu $m = 3$ ir $f(x_1, x_2, x_3) = x_1 + x_3$. Prisiminkime, kad skaičiuojama virš kūno $\mathbb{F}_2$. Tada

$$\begin{aligned} &\Delta_1 f = f(x_1 + 1, x_2, x_3) + f(x_1, x_2, x_3) = (x_1 + 1 + x_3) + (x_1 + x_3) = 1, \\ &\Delta_2 f = f(x_1, x_2 + 1, x_3) + f(x_1, x_2, x_3) = (x_1 + x_3) + (x_1 + x_3) = 0, \\ &\Delta_3 f = f(x_1, x_2, x_3 + 1) + f(x_1, x_2, x_3) = 1. \end{aligned}$$

Tegu dabar $f(x_1, x_2, x_3) = x_1x_2 + x_3$. Gauname, kad

$$\begin{aligned} &\Delta_1 f = f(x_1 + 1, x_2, x_3) + f(x_1, x_2, x_3) = ((x_1 + 1)x_2 + x_3) + (x_1x_2 + x_3) = x_2, \\ &\Delta_2 f = f(x_1, x_2 + 1, x_3) + f(x_1, x_2, x_3) = (x_1(x_2 + 1) + x_3) + (x_1x_2 + x_3) = x_1, \\ &\Delta_3 f = f(x_1, x_2, x_3 + 1) + f(x_1, x_2, x_3) = 1. \end{aligned}$$

□



**3.2.13 teiginys.** *Jei $f$ yra afinioji loginė funkcija, tai $\Delta_i f$ yra konstanta. Tiksliau, jei*

$$f(x_1, \ldots, x_m) = \sum_{i=1}^{m} \lambda_i x_i + \mu,$$

*tai*

$$\Delta_i f(x) = \lambda_i.$$

**3.2.14 užduotis.** Įrodyti teiginį.

Taigi, $\overrightarrow{\Delta_i f} = \vec{\lambda}_i = \lambda_i\,(1, 1, \ldots, 1)$.

Tarkime, kad į kanalą pasiuntėme kodo $RM(1, m)$ žodį $\vec{f}$, atitinkantį afiniąją loginę funkciją $f(x) = \sum_{i=1}^{m} \lambda_i x_i + \mu$. Tarkime, kad iš kanalo gavome vektorių $\vec{g}$, atitinkantį (nebūtinai afiniąją) loginę funkciją $g$. Žinome, kad $\vec{g} = \vec{f} + \vec{e}$, kur $\vec{e}$ — klaidų vektorius, atitinkantis loginę funkciją $e$. Kadangi $\vec{f} + \vec{e} = \overrightarrow{f + e} = \vec{g}$, tai $g = f + e$.

Padarykime tokią *prielaidą*: klaidų skaičius vektoriuje mažesnis už ketvirtadalį vektoriaus ilgio, t. y. $w(\vec{e}) < \frac{2^m}{4} = 2^{m-2}$. Parodysime, kad tokiu atveju galima ištaisyti visas klaidas.

Tegu $1 \leqslant i \leqslant m$. Tada

$$\begin{aligned} \Delta_i g &= g(x + a_i) + g(x) \\ &= f(x + a_i) + e(x + a_i) + f(x) + e(x) \\ &= \Delta_i f + \Delta_i e \\ &= \lambda_i + \Delta_i e, \end{aligned}$$

todėl

$$\overrightarrow{\Delta_i g} = \lambda_i\,(1, \ldots, 1) + \overrightarrow{\Delta_i e}. \tag{3.2}$$

Įvertinkime vektoriaus $\overrightarrow{\Delta_i e}$ svorį $w(\overrightarrow{\Delta_i e})$. Tai leis įvertinti vektoriaus $\overrightarrow{\Delta_i g}$ svorį.

Visų pirma pastebėkime, kad kai $x = (x_1, x_2, \ldots, x_m)$ perbėga visų galimų tokių vektorių erdvę $\mathbb{F}_2^m$, tai $x + a_i$ taip pat perbėga visą šią erdvę $\mathbb{F}_2^m$, todėl atvaizdis $\mathbb{F}_2^m \to \mathbb{F}_2^m$, $x \mapsto x + a_i$ yra bijekcija (keitinys, perstata).

Kadangi skaičiuojant vektoriaus $\overrightarrow{e(x)}$ reikšmes $x$ perbėga visą erdvę $\mathbb{F}_2^m$, tai skaičiuojant $\overrightarrow{e(x + a_i)}$ gauname tas pačias reikšmes, tik išrikiuotas kita tvarka.

**3.2.15 pavyzdys.** Tegu $m = 3$, $a_2 = (0, 1, 0)$ ir $\overrightarrow{e(x)} = (e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7)$. Pateiktoje lentelėje matome, kad, kai $x$ perbėga $\mathbb{F}_2^3$, $x + a_2$ irgi perbėga $\mathbb{F}_2^3$. Todėl $e(x + a_2)$ įgyja tas pačias reikšmes, kaip $e(x)$, tik kita eilės tvarka. Pavyzdžiui, jei $x = 101$, tai $e(x + a_2) = e(101 + 010) = e(111) = e_7$, ir pan.

| $x$ | $x + a_2$ | $e(x)$ | $e(x + a_2)$ |
|---|---|---|---|
| 000 | 010 | $e_0$ | $e_2$ |
| 001 | 011 | $e_1$ | $e_3$ |
| 010 | 000 | $e_2$ | $e_0$ |
| 011 | 001 | $e_3$ | $e_1$ |
| 100 | 110 | $e_4$ | $e_6$ |
| 101 | 111 | $e_5$ | $e_7$ |
| 110 | 100 | $e_6$ | $e_4$ |
| 111 | 101 | $e_7$ | $e_5$ |

□

Taigi, vektorius $\overrightarrow{e(x+a_i)}$ ir $\overrightarrow{e(x)}$ sudaro tie patys elementai, tik skirtingai išrikiuoti, todėl jų svoriai sutampa: $w(\overrightarrow{e(x+a_i)}) = w(\overrightarrow{e(x)})$.

Pasinaudoję trikampio nelygybe ir mūsų prielaida, gauname:

$$w(\overrightarrow{\Delta_i e}) = w(\overrightarrow{e(x+a_i)} + \overrightarrow{e(x)}) \leqslant w(\overrightarrow{e(x+a_i)}) + w(\overrightarrow{e(x)}) = 2w(\overrightarrow{e(x)}) < 2 \cdot 2^{m-2} = 2^{m-1}.$$

Kadangi vektoriaus $\overrightarrow{\Delta_i e}$ ilgis yra $2^m$, tai matome, kad vienetai stovi mažiau nei pusėje jo pozicijų. Grįžkime prie (3.2) lygybės.

- Jei $\lambda_i = 0$, tai $\overrightarrow{\Delta_i g} = \overrightarrow{\Delta_i e}$, todėl $w(\overrightarrow{\Delta_i g}) = w(\overrightarrow{\Delta_i e}) < 2^{m-1}$.
- Jei $\lambda_i = 1$, tai

$$w(\overrightarrow{\Delta_i g}) = w((1,1,\ldots,1) + \overrightarrow{\Delta_i e}) = 2^m - w(\overrightarrow{\Delta_i e}) > 2^m - 2^{m-1} = 2^{m-1}(2-1) = 2^{m-1}.$$

Tai mums leidžia nustatyti $\lambda_i$ kiekvienam $i = 1, \ldots, m$ tokiu būdu. Gavę iš kanalo vektorių $\vec{g}$ apskaičiuojame vektoriaus $\overrightarrow{\Delta_i g}$ svorį $w(\overrightarrow{\Delta_i g})$.

- Jei $w(\overrightarrow{\Delta_i g}) < 2^{m-1}$, tai padarome išvadą, kad $\lambda_i = 0$.
- Jei $w(\overrightarrow{\Delta_i g}) > 2^{m-1}$, tai padarome išvadą, kad $\lambda_i = 1$.

Belieka nustatyti $\mu$. Pažymėkime

$$h(x) = g(x) + \sum_{i=1}^{m} \lambda_i x_i.$$

Tada

$$h(x) = f(x) + e(x) + \sum_{i=1}^{m} \lambda_i x_i = \sum_{i=1}^{m} \lambda_i x_i + \mu + e(x) + \sum_{i=1}^{m} \lambda_i x_i = \mu + e(x).$$

Todėl

$$\overrightarrow{h(x)} = \mu\,(1,1,\ldots,1) + \overrightarrow{e(x)}.$$

Vėl gauname, kad:

- jei $\mu = 0$, tai $w(\overrightarrow{h(x)}) = w(\overrightarrow{e(x)}) < 2^{m-2}$,
- jei $\mu = 1$, tai $w(\overrightarrow{h(x)}) = 2^m - w(\overrightarrow{e(x)}) > 2^m - 2^{m-2} = 2^{m-2}(4-1) = 3 \cdot 2^{m-2}$.

Taigi, radę visus $\lambda_i$, apskaičiuojame vektoriaus

$$\overrightarrow{h(x)} = \overrightarrow{g(x)} + \overrightarrow{\sum_{i=1}^{m} \lambda_i x_i}$$

svorį $w(\overrightarrow{h(x)})$, ir:

- jei $w(\overrightarrow{h(x)}) < 2^{m-2}$, tai nusprendžiame, kad $\mu = 0$,



- jei $w(\overrightarrow{h(x)}) > 3 \cdot 2^{m-2}$, tai $\mu = 1$.

Jei $w(\overrightarrow{\Delta_i g})$ ar $w(\overrightarrow{h(x)})$ netenkina tų sąlygų, tai reiškia, kad klaidų kanale buvo padaryta daugiau, nei numatyta prielaidoje, ir šis algoritmas jų nebegali ištaisyti.

**3.2.16 pavyzdys.** Tegu $m = 3$. Dekodavimo algoritmas taisys mažiau, nei $2^{m-2} = 2^{3-2} = 2$ klaidas, t. y. galės ištaisyti tik vieną klaidą.

Tarkime, pranešimas yra $(\lambda_1, \lambda_2, \lambda_3, \mu) = (1,0,1,1)$. Jį reikia užkoduoti. Užkoduotas vektorius bus vektorius $\vec{f}$, atitinkantis loginę funkciją $f(x) = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \mu = x_1 + x_3 + 1$. Sudarome funkcijos $f$ reikšmių lentelę ir gauname, kad $\vec{f} = (1,0,1,0,0,1,0,1)$. Šį kodo $RM(1,3)$ žodį ir siunčiame į kanalą.

Tarkime, kanale buvo padaryta viena klaida. Tarkime, klaida buvo padaryta ketvirtojoje pozicijoje, t. y. klaidų vektorius yra $\vec{e} = (0,0,0,1,0,0,0,0)$. Klaidų vektorių atitinkančios funkcijos rasti nereikia, bet įdomumo dėlei pastebėkime, kad tai funkcija $e(x_1, x_2, x_3) = (1 + x_1)x_2x_3$, nes iš vektoriaus $\vec{e}$ matome, kad funkcija $e$ įgyja reikšmę 1 tik su kintamųjų reikšmių rinkiniu $(0,1,1)$ (prisiminkime iš diskrečiosios matematikos kurso, kad gauti loginės funkcijos išraišką turėdami jos teisingumo reikšmių lentelę galime sudarydami jos normaliąsias formas — normaliąją konjunkcinę ar disjunkcinę formas).

Iš kanalo gauname vektorių $\vec{g} = \vec{f} + \vec{e} = (1,0,1,1,0,1,0,1)$. Dekoduodami nustatysime $\lambda_1$, $\lambda_2$, $\lambda_3$ ir $\mu$ reikšmes, t. y. surasime pradinį pranešimą $(\lambda_1, \lambda_2, \lambda_3, \mu)$.

Pradėsime nuo $\lambda_1$ radimo. Reikia apskaičiuoti $\overrightarrow{\Delta_1 g} = \overrightarrow{g(x)} + \overrightarrow{g(x+a_1)}$. Kam lygus $\overrightarrow{g(x+a_1)}$? Kaip ir 3.2.15 pavyzdyje sudarome lentelę, pagal kurią nustatome, kokiu būdu sukeisti vektoriaus $\overrightarrow{g(x)}$ koordinates, kad gautume vektorių $\overrightarrow{g(x+a_1)}$.

| $x$ | $x+a_1$ | $g(x)$ | $g(x+a_1)$ | $x+a_2$ | $g(x+a_2)$ | $x+a_3$ | $g(x+a_3)$ |
|---|---|---|---|---|---|---|---|
| 000 | 100 | $g_0$ | $g_4$ | 010 | $g_2$ | 001 | $g_1$ |
| 001 | 101 | $g_1$ | $g_5$ | 011 | $g_3$ | 000 | $g_0$ |
| 010 | 110 | $g_2$ | $g_6$ | 000 | $g_0$ | 011 | $g_3$ |
| 011 | 111 | $g_3$ | $g_7$ | 001 | $g_1$ | 010 | $g_2$ |
| 100 | 000 | $g_4$ | $g_0$ | 110 | $g_6$ | 101 | $g_5$ |
| 101 | 001 | $g_5$ | $g_1$ | 111 | $g_7$ | 100 | $g_4$ |
| 110 | 010 | $g_6$ | $g_2$ | 100 | $g_4$ | 111 | $g_7$ |
| 111 | 011 | $g_7$ | $g_3$ | 101 | $g_5$ | 110 | $g_6$ |

Matome, kad jei

$$\overrightarrow{g(x)} = (g_0, g_1, \ldots, g_7) = (1,0,1,1,0,1,0,1),$$

tai

$$\overrightarrow{g(x+a_1)} = (g_4, g_5, g_6, g_7, g_0, g_1, g_2, g_3) = (0,1,0,1,1,0,1,1).$$

Todėl

$$\overrightarrow{\Delta_1 g} = \overrightarrow{g(x)} + \overrightarrow{g(x+a_1)} = (1,1,1,0,1,1,1,0).$$

Taigi, $w(\overrightarrow{\Delta_1 g}) = 6 > 2^{m-1} = 4$, todėl nusprendžiame, kad $\lambda_1 = 1$.

Analogiškai

$$\overrightarrow{g(x+a_2)} = (g_2, g_3, g_0, g_1, g_6, g_7, g_4, g_5) = (1,1,1,0,0,1,0,1),$$

todėl

$$\overrightarrow{\Delta_2 g} = \overrightarrow{g(x)} + \overrightarrow{g(x+a_2)} = (0,1,0,1,0,0,0,0).$$

Taigi, $w(\overrightarrow{\Delta_2 g}) = 2 < 4$, todėl nusprendžiame, kad $\lambda_2 = 0$.

Taip pat ir

$$\overrightarrow{g(x+a_3)} = (g_1, g_0, g_3, g_2, g_5, g_4, g_7, g_6) = (0,1,1,1,1,0,1,0),$$

todėl

$$\overrightarrow{\Delta_3 g} = \overrightarrow{g(x)} + \overrightarrow{g(x+a_3)} = (1,1,0,0,1,1,1,1).$$

Taigi, $w(\overrightarrow{\Delta_3 g}) = 6 > 4$, todėl nusprendžiame, kad $\lambda_3 = 1$.

Liko rasti $\mu$. Sudarome funkciją $f'(x) = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 = x_1 + x_3$, randame ją atitinkantį vektorių $\overrightarrow{f'(x)} = (0,1,0,1,1,0,1,0)$. Tada

$$\overrightarrow{h(x)} = \overrightarrow{g(x)} + \overrightarrow{f'(x)} = (1,1,1,0,1,1,1,1).$$

Kadangi $w(\overrightarrow{h(x)}) = 7 > 3 \cdot 2^{m-2} = 3 \cdot 2 = 6$, tai nusprendžiame, kad $\mu = 1$.

Taigi, dekodavę gauname, kad pradinis pranešimas buvo $(\lambda_1, \lambda_2, \lambda_3, \mu) = (1,0,1,1)$, t. y. padaryta klaida buvo ištaisyta. Įdomumo dėlei dar galime rasti klaidų vektorių

$$\overrightarrow{e(x)} = \overrightarrow{h(x)} + \mu\,(1,1,\ldots,1) = (0,0,0,1,0,0,0,0)$$

ir išsiųstą kodo žodį $\vec{f} = \vec{g} - \vec{e} = (1,0,1,0,0,1,0,1)$. □

### 3.2.3 Minimalus atstumas

**3.2.17 teiginys.** *Kodo $RM(1,m)$ minimalus atstumas yra $2^{m-1}$.*

Be įrodymo.

## 3.3 Naujų kodų sudarymo būdai

Iš jau turimų kodų galima sudaryti naujus, stengiantis pagerinti jų parametrus.

### 3.3.1 Plėtinys

**3.3.1 apibrėžimas.** *Tegu $C$ yra ilgio $n$ kodas (nebūtinai tiesinis) virš $\mathbb{F}_q$. Jo* plėtiniu *vadinsime kodą*

$$C^+ = \{(x_1, x_2, \ldots, x_n, x_{n+1}) \in \mathbb{F}_q^{n+1} : (x_1, x_2, \ldots, x_n) \in C, \sum_{j=1}^{n+1} x_j = 0\}.$$

Taigi, plėtinys yra kodas, gaunamas tiesiog prie kiekvieno kodo $C$ žodžio prijungus po tokį simbolį $x_{n+1}$, kad gauto žodžio koordinačių suma būtų lygi nuliui.

**3.3.2 pavyzdys.** Jei $C = \{1200, 1001, 0112\}$ yra kodas virš $\mathbb{F}_3$, tai $C^+ = \{12000, 10011, 01122\}$. Jei $C' = \{1100, 1000, 1110\}$ yra dvejetainis kodas, tai $C'^+ = \{11000, 10001, 11101\}$. □



Matome, kad plėtinio sudarymas kai kuriais atvejais leidžia labai paprastai padidinti kodo minimalų atstumą (3.3.2 pavyzdyje kodo $C$ minimalus atstumas $d = 2$, o jo plėtinio $C^+$ minimalus atstumas $d^+ = 3$).

**3.3.3 teiginys.** *Jei $C$ yra tiesinis kodas, generuotas matricos $G$, tai jo plėtinys $C^+$ irgi yra tiesinis kodas, ir jo generuojanti matrica yra $G^+ = (G|b)$, kur $b$ yra toks vektorius-stulpelis, kad kiekvienos $G^+$ eilutės elementų suma lygi* 0.

Įrodyti savarankiškai.

**3.3.4 pavyzdys.** Jei dvejetainis kodas $C$ yra generuotas matricos

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

tai jo plėtinio $C^+$ generuojanti matrica yra

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

**3.3.5 užduotys.** 1. Įrodykite tokius teiginius.

a) Jei $C$ yra $(n, M, d)$ kodas, tai jo plėtinys $C^+$ yra $(n+1, M, d^+)$ kodas, kur $d \leqslant d^+ \leqslant d+1$.

b) Jei $C$ yra dvejetainis kodas, kurio minimalus atstumas yra $d$, tai plėtinio minimalus atstumas

$$d^+ = \begin{cases} d, & \text{jei } d \text{ — lyginis,} \\ d+1, & \text{jei } d \text{ — nelyginis.} \end{cases}$$

c) Jei tiesinio kodo $C$ kontrolinė matrica yra $H$, tai plėtinio $C^+$ kontrolinė matrica yra

$$H^+ = \left(\begin{array}{ccc|c} & & & 0 \\ & H & & \vdots \\ & & & 0 \\ \hline 1 & \cdots & 1 & 1 \end{array}\right).$$

2. Matome, kad plėtinio minimalus atstumas $d^+$ gali būti vienetu didesnis, negu pradinio kodo minimalus atstumas $d$. Bet sunku pasakyti, ar plėtinio minimalus atstumas bus didesnis už kodo minimalų atstumą, ar ne. Pavyzdžiui, jei kodai $C_1$ ir $C_2$ virš $\mathbb{F}_3$ yra generuoti atitinkamai matricų

$$G_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \text{ ir } G_2 = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 2 \end{pmatrix},$$

tai plėtinio $C_1^+$ minimalus atstumas didesnis, nei kodo $C_1$, o plėtinio $C_2^+$ — ne. Iš tiesų, įsitikinkite, kad kodų $C_1, C_2, C_2^+$ minimalus atstumas yra 2, o kodo $C_1^+$ — 3.

### 3.3.2 Sutrumpintas kodas

**3.3.6 apibrėžimas.** *Tegu $C$ yra ilgio $n$ kodas virš $\mathbb{F}_q$, $J$ yra indeksų aibės $\{1,\dots,n\}$ poaibis. Kodo $C$* aibėje $J$ sutrumpintu kodu *vadinsime kodą*

$$C_J = \{(x_i)_{i\in\{1,\dots,n\}\setminus J} \in \mathbb{F}_q^{n-|J|} : (x_i)_{i\in\{1,\dots,n\}} \in C \text{ su kuriais nors } x_i \in \mathbb{F}_q, i \in J\}.$$

*Kai nenurodome, kokioje aibėje trumpiname, kodo $C$* sutrumpintu kodu *vadiname kodą $C_{\{n\}}$.*

Taigi, kodo $C$ aibėje $J$ sutrumpintas kodas gaunamas tiesiog iš kiekvieno kodo $C$ žodžio pašalinus koordinates, priklausančias aibei $J$. Gauname ilgio $n-|J|$ kodą. Jei nenurodoma, kokioje aibėje trumpiname, tai šaliname paskutinę ($n$-tąją) koordinatę, gauname $n-1$ ilgio kodą.

**3.3.7 pavyzdys.** Jei $C = \{120020, 100121, 011201\}$ yra kodas virš $\mathbb{F}_3$, tai jo aibėje $\{2,4,6\}$ sutrumpintas kodas gaunamas iš kiekvieno kodo $C$ žodžio pašalinus antrą, ketvirtą ir šeštą koordinates: $C_{\{2,4,6\}} = \{102, 010\}$. Kadangi sutrumpinę žodžius 120020 ir 100121 gauname tą patį žodį 102, tai sutrumpintame kode žodžių bus mažiau, nei pradiniame. Kodo $C$ sutrumpintas kodas gaunamas iš kiekvieno kodo $C$ žodžio pašalinus paskutinę koordinatę: $C_{\{6\}} = \{12002, 10012, 01120\}$. □

**3.3.8 teiginys.** *Jei kodas $C$ yra tiesinis, tai jo aibėje $J$ sutrumpintas kodas irgi yra tiesinis, ir jo generuojanti matrica gaunama iš kodo $C$ generuojančios matricos pašalinus stulpelius, kurių numeriai priklauso aibei $J$.*

**3.3.9 pavyzdys.** Jei dvejetainis kodas $C$ yra generuotas matricos

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

tai jo aibėje $\{1,4,5\}$ sutrumpinto kodo generuojančią matricą $G'$ gausime, pašalinę iš matricos $G$ pirmą, ketvirtą ir penktą stulpelius:

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

Kodo $C$ sutrumpinto kodo generuojančią matricą $G'$ gausime, pašalinę iš matricos $G$ paskutinį stulpelį:

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

Aibėje $J$ sutrumpintų kodų dimensija gali sumažėti, dėl to reikia patikrinti, ar pašalinę generuojančios matricos stulpelius negausime tiesiškai priklausomų eilučių. Jei gauname, tai paliekame tik maksimalią tiesiškai nepriklausomų eilučių aibę, likusias eilutes pašalindami.

**3.3.10 pavyzdys.** Tarkime, dvejetainis kodas $C$ yra generuotas matricos

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



Pašalinę paskutinį jos stulpelį, gauname matricą

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

kurios eilutės tiesiškai priklausomos (pirmos dvi eilutės lygios). Pašaliname vieną iš tų lygių eilučių ir gauname sutrumpinto kodo generuojančią matricą

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

**3.3.11 užduotys.** Įrodykite tokius teiginius.

1. Jei $C$ yra $(n, M, d)$ kodas virš $\mathbb{F}_q$, tai jo sutrumpintas kodas $C'$ yra $(n-1, M', d')$ kodas virš $\mathbb{F}_q$, kur $d-1 \leqslant d' \leqslant d$, $M/q \leqslant M' \leqslant M$. Jei $d \geqslant 2$, tai $M' = M$.

2. Tegu $C$ yra tiesinis $[n,k,d]$ kodas virš $\mathbb{F}_q$, kurio kontrolinė matrica yra $H$. Tada jo sutrumpintas kodas $C'$ yra tiesinis $[n-1, k', d']$ kodas, kur $d-1 \leqslant d' \leqslant d$, $k-1 \leqslant k' \leqslant k$. Jei $d \geqslant 2$, tai $k' = k$. Kodo $C'$ kontrolinė matrica $H'$ gaunama taip: jei matricos $H$ paskutinis stulpelis yra nulinis, $H'$ gaunama iš $H$ pašalinant paskutinį stulpelį, o jei ne, tai reikia suvesti $H$ į tokį pavidalą, kad tik vienos eilutės paskutinėje pozicijoje būtų nenulinis elementas, ir pašalinti tą eilutę bei paskutinį stulpelį iš $H$.

### 3.3.3 Sumažintas kodas

**3.3.12 apibrėžimas.** *Tegu $C$ yra ilgio $n$ kodas virš $\mathbb{F}_q$, $J$ yra indeksų aibės $\{1,\dots,n\}$ poaibis. Kodo $C$* aibėje $J$ sumažintu kodu *vadinsime kodą*

$$C_{\setminus J} = \{(x_i)_{i\in\{1,\dots,n\}\setminus J} \in \mathbb{F}_q^{n-|J|} : (x_i)_{i\in\{1,\dots,n\}} \in C \text{ toks, kad } x_i = 0\ \forall i \in J\}.$$

*Kai nenurodome, kokioje aibėje mažiname, kodo $C$* sumažintu kodu *vadiname kodą $C_{\setminus\{n\}}$.*

Taigi, kodo $C$ aibėje $J$ sumažintą kodą gauname, išrinkę kodo $C$ žodžius, kurių koordinatėse, priklausančiose aibei $J$, yra nuliai, ir pašalinę tas koordinates. Gauname ilgio $n-|J|$ kodą. Jei nenurodoma, kokioje aibėje mažiname, tai šaliname paskutinę ($n$-tąją) koordinatę iš tų kodo $C$ žodžių, kurių paskutinė koordinatė lygi nuliui. Gauname $n-1$ ilgio kodą.

**3.3.13 pavyzdys.** Jei $C = \{120020, 100120, 011200, 210222\}$ yra kodas virš $\mathbb{F}_3$, tai jo aibėje $\{3,6\}$ sumažintas kodas gaunamas taip. Visų pirma išrenkame kodo $C$ žodžius, kurių trečioje ir šeštoje vietoje yra nuliai. Tokių žodžių aibė bus $\{120020, 100120\}$. Tada iš jų pašaliname trečią ir šeštą koordinates: $C_{\setminus\{3,6\}} = \{1202, 1012\}$. Kodo $C$ sumažintas kodas bus $C_{\setminus\{6\}} = \{12002, 10012, 01120\}$. □

**3.3.14 teiginys.** *Jei kodas $C$ yra tiesinis, tai jo aibėje $J$ sumažintas kodas irgi yra tiesinis, ir jo kontrolinė matrica gaunama iš kodo $C$ kontrolinės matricos pašalinus stulpelius, kurių numeriai priklauso aibei $J$.*

Taigi, dabar šaliname ne generuojančios matricos, kaip trumpindami kodą, o kontrolinės matricos stulpelius. Kaip šalinti stulpelius, matėme 3.3.9 pavyzdyje, tik jame dabar žodžius „generuojanti matrica“ reikėtų pakeisti į „kontrolinė matrica“.

Kaip ir sutrumpintiems kodams, galime gauti, kad pašalinus stulpelius matricoje atsiras tiesiškai priklausomų eilučių. Tokiu atveju paliekame tik maksimalią tiesiškai nepriklausomų eilučių aibę, likusias eilutes pašalindami.

**3.3.15 užduotys.** Įrodykite tokius teiginius.

1. Jei $C$ yra $(n, M, d)$ kodas virš $\mathbb{F}_q$, tai jo sumažintas kodas $C'$ yra $(n-1, M', d')$ kodas virš $\mathbb{F}_q$, kur $0 \leqslant M' \leqslant M$, $d' \geqslant d$.

2. Tegu $C$ yra tiesinis $[n, k, d]$ kodas virš $\mathbb{F}_q$. Jo sumažintas kodas $C'$ taip pat yra tiesinis kodas. Jei kodo $C$ žodžių paskutinėje pozicijoje yra tik nuliai, tai $C'$ yra $[n-1, k, d]$ kodas. Priešingu atveju $C'$ yra $[n-1, k-1, d']$ kodas, kur $d' \geqslant d$.



# 4 skyrius
# Cikliniai kodai

Ruošiant šį skyrių naudotasi V.Stakėno knygute „Informacijos kodavimas“, VU leidykla, 1996.

## 4.1 Vektorių ir polinomų atitiktis

Nagrinėsime žodžius virš $\mathbb{F}_q$. Juos sudėti jau mokame. Dabar bandysime apibrėžti jų sandaugos operaciją.

Tegu $n \geqslant 1$. Pažymėkime visų polinomų virš $\mathbb{F}_q$ aibę $\mathbb{F}_q[x]$, o visų polinomų virš $\mathbb{F}_q$, kurių laipsnis mažesnis už $n$, aibę $\mathbb{F}_{q,n}[x]$:

$$\begin{aligned} \mathbb{F}_q[x] &= \{a_0 + a_1 x + \cdots + a_{m-1}x^{m-1} : a_i \in \mathbb{F}_q \ \forall i, m \geqslant 1\}, \\ \mathbb{F}_{q,n}[x] &= \{a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} : a_i \in \mathbb{F}_q \ \forall i\}. \end{aligned}$$

Taigi, $\mathbb{F}_q[x]$ yra visų polinomų virš $\mathbb{F}_q$ aibė, o $\mathbb{F}_{q,n}[x]$ — aibė polinomų virš $\mathbb{F}_q$, kurių laipsnis mažesnis už $n$. Nesunku įsitikinti, kad $\mathbb{F}_q[x]$ ir $\mathbb{F}_{q,n}[x]$ yra tiesinės erdvės virš $\mathbb{F}_q$.

Apibrėšime abipusiškai vienareikšmę žodžių iš $\mathbb{F}_q^n$ ir polinomų iš $\mathbb{F}_{q,n}[x]$ atitiktį:

$$(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{F}_q^n \quad \longleftrightarrow \quad a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} \in \mathbb{F}_{q,n}[x].$$

Nesunku įsitikinti, kad ši atitiktis iš tikrųjų yra tiesinių erdvių $\mathbb{F}_q^n$ ir $\mathbb{F}_{q,n}[x]$ izomorfizmas. Taigi, žodžius iš $\mathbb{F}_q^n$ galime tapatinti su atitinkamais $\mathbb{F}_{q,n}[x]$ polinomais.

**4.1.1 pavyzdys.** Tegu $q = 5$, $n = 7$. Tada žodį $(2, 4, 0, 1, 3, 3, 4) \in \mathbb{F}_5^7$ atitinka polinomas $2 + 4x + x^3 + 3x^4 + 3x^5 + 4x^6 \in \mathbb{F}_{5,7}[x]$. □

Prisiminkime iš algebros kurso, kad žiedas $A$ — tai algebrinė struktūra, apibrėžiama taip pat, kaip kūnas (žr. 2.1.1 poskyrį), išskyrus tai, kad nėra reikalaujama, kad kiekvienas $A$ elementas turėtų atvirkštinį elementą daugybos atžvilgiu (elemento $a \in A$ atvirkštiniu elementu vadiname tokį elementą $\bar{a} \in A$, kad $a\bar{a} = 1$). Pavyzdžiui, sveikųjų skaičių aibė $\mathbb{Z}$ yra žiedas. Tai nėra kūnas, nes, pavyzdžiui, 2 neturi atvirkštinio elemento, t. y. nėra tokio sveikojo skaičiaus, kurį sudauginę su 2 gautume 1.

Nesunku patikrinti, kad $\langle \mathbb{F}_q[x], +, \times \rangle$ yra žiedas. Jame veikia dalybos su liekana algoritmas:

**4.1.2 teorema.** *Bet kokiems polinomams $f(x), g(x)$ egzistuoja tokie vieninteliai polinomai $d(x), r(x)$, kad*

$$f(x) = d(x) \times g(x) + r(x), \ \textit{kur} \ \deg r < \deg g.$$

Čia $\deg r$ žymi polinomo $r$ laipsnį. Polinomai $d(x), r(x)$ vadinami polinomo $f(x)$ dalybos iš $g(x)$ atitinkamai *dalmeniu* ir *liekana*.

Praktikoje dalinti polinomus galime naudodami metodą, analogišką sveikųjų skaičių dalybai — „dalybą kampu“.

**4.1.3 pavyzdys.** Padalinkime, pavyzdžiui, $f(x) = x^8 - 1$ iš $g(x) = x^3 + 2x^2 + 2$, kur $f(x), g(x) \in \mathbb{F}_3[x]$. Dalyba kampu parodyta 4.1 paveiksle. Matome, kad gavome dalmenį $d(x) = x^5 + x^4 + x^3 + 2x^2 + 1$ ir liekaną $r(x) = 0$. Nustojame dalinti, kai gauname liekaną, kurios laipsnis mažesnis už $\deg g$. □

$$
\begin{array}{rrrrrrrrl}
 & x^8 & & & & & & -1 & |x^3+2x^2+2 \\
\cline{9-9}
 & x^8 & +2x^7 & & +2x^5 & & & & x^5+x^4+x^3+2x^2+1 \\
\cline{1-8}
- & & x^7 & & +x^5 & & & +2 & \\
\cline{2-8}
 & - & x^7 & +2x^6 & & +2x^4 & & & \\
\cline{3-8}
 & & & x^6 & +x^5 & +x^4 & & +2 & \\
\cline{3-8}
 & & - & x^6 & +2x^5 & & +2x^3 & & \\
\cline{4-8}
 & & & & 2x^5 & +x^4 & +x^3 & +2 & \\
\cline{4-8}
 & & & - & 2x^5 & +x^4 & & +x^2 & \\
\cline{5-8}
 & & & & & & x^3+2x^2 & +2 & \\
\cline{7-8}
 & & & & & - & x^3+2x^2 & +2 & \\
\cline{7-8}
 & & & & & & & 0 &
\end{array}
$$

4.1 pav.: Polinomų $x^8-1$ ir $x^3+2x^2+2$ dalyba kampu

Į $\mathbb{F}_{q,n}[x]$ žiūrime kaip į liekanų, gautų dalijant polinomus iš $x^n-1$, aibę (liekanų laipsniai neviršys $n-1$).

**4.1.4 pastaba.** Pastebėkime, kad, kadangi $x^n=(x^n-1)+1$, tai dalindami $x^n$ iš $x^n-1$ gausime liekaną 1. Toliau, kadangi $x^{n+1}=x^nx=(x^n-1)x+x$, tai dalindami $x^{n+1}$ iš $x^n-1$ gausime liekaną $x$, ir t.t. Apskritai galime laikyti, kad skaičiuodami liekaną galime $x^n$ pakeisti 1. Tuo galime pasinaudoti, norėdami nesunkiai rasti polinomo dalybos iš $x^n-1$ liekaną.

**4.1.5 pavyzdys.** Kokia bus dvejetainio polinomo $x^8+x^6+x^5+x^3+x^2+1$ dalybos iš $x^5-1$ liekana? Pasinaudojame tuo, kad $x^8$ dalybos iš $x^5-1$ liekana yra $x^3$ (nes $x^8=x^5x^3$, ir $x^5$ galime pakeisti 1). Analogiškai vietoj $x^6$ gauname $x$ ir vietoj $x^5$ gauname 1. Taigi, liekana bus $x^3+x+1+x^3+x^2+1=x^2+x$. □

Apibrėžkime polinomų $p(x),q(x)\in\mathbb{F}_{q,n}[x]$ daugybą aibėje $\mathbb{F}_{q,n}[x]$. Sudauginkime $p(x)$ ir $q(x)$ kaip įprastus polinomus. Aišku, sandaugos laipsnis gali viršyti $n-1$. Kad sumažintume laipsnį iki $n-1$, padalinkime sandaugą iš $x^n-1$ ir imkime liekaną $r(x)$. Jos laipsnis bus mažesnis už $n$. Ją ir laikysime polinomų $p(x)$ ir $q(x)$ sandauga. Reziumuojant: $p(x)q(x)=r(x)$, kur $r(x)$ yra polinomų $p(x)\times q(x)$ ir $x^n-1$ dalybos liekana.

**4.1.6 pavyzdys.** Sudauginkime dvejetainius polinomus $x^2+1$ ir $x^3+x+1$ aibėje $\mathbb{F}_{2,5}[x]$. Tam juos sudauginame, naudodami įprastą polinomų daugybą $\times$ aibėje $\mathbb{F}_2[x]$:

$$(x^2+1)\times(x^3+x+1)=x^5+x^2+x+1.$$

Tada skaičiuojame dalybos iš $x^5-1$ liekaną kaip 4.1.5 pavyzdyje ir gauname $x^2+x$. Taigi, aibėje $\mathbb{F}_{2,5}[x]$

$$(x^2+1)(x^3+x+1)=x^2+x.$$ □

Šitaip aibėje $\mathbb{F}_{q,n}[x]$ apibrėžę polinomų daugybą, gauname žiedą $\langle\mathbb{F}_{q,n}[x],+,\cdot\rangle$.



Iš bet kokios lygybės žiede $\langle\mathbb{F}_q[x],+,\times\rangle$ gauname lygybę žiede $\langle\mathbb{F}_{q,n}[x],+,\cdot\rangle$, polinomus pakeitę dalybos iš $x^n-1$ liekanomis, o įprastą polinomų daugybą '$\times$' keisdami žiedo $\langle\mathbb{F}_{q,n}[x],+,\cdot\rangle$ daugyba '$\cdot$'.

Nuo šiol abi daugybos operacijas žymėsime vienodai — '$\cdot$'. Apie kurią daugybos operaciją šnekama, bus aišku iš konteksto.

Dar vienas priminimas iš algebros: žiedo požiedžiu vadinamas toks jo netuščias poaibis, kuris pats yra žiedas (tenkina žiedo apibrėžimą).

**4.1.7 apibrėžimas.** *Žiedo $\mathbb{F}_{q,n}[x]$ požiedį $I$ vadinsime* idealu, *jeigu patenkinta tokia sąlyga:*

$$\forall q(x)\in I\quad xq(x)\in I.$$

Iš apibrėžimo išplaukia, kad visiems $r(x)\in\mathbb{F}_{q,n}[x]$ ir $q(x)\in I$, $r(x)q(x)\in I$.

**4.1.8 apibrėžimas.** *Idealą*

$$\langle q(x)\rangle=\{r(x)q(x):r(x)\in\mathbb{F}_{q,n}[x]\}$$

*vadiname* pagrindiniu idealu, generuotu polinomo $q(x)$, *o pats $q(x)$ vadinamas idealo $\langle q(x)\rangle$* generatoriumi.

Nesunku įrodyti, kad visi žiedo $\mathbb{F}_{q,n}[x]$ idealai yra pagrindiniai idealai (t. y. visi turi generatorių).

## 4.2 Cikliniai kodai

### 4.2.1 Apibrėžimas

**4.2.1 apibrėžimas.** *Tiesinį kodą $C\subset\mathbb{F}_q^n$ vadinsime* cikliniu, *jei iš $(c_0,c_1,\dots,c_{n-2},c_{n-1})\in C$ išplaukia, kad $(c_{n-1},c_0,\dots,c_{n-2})\in C$.*

**4.2.2 pavyzdys.** Tarkime, $C$ yra dvejetainis tiesinis kodas, generuotas matricos

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

Tada

$$C=\{0000,1001,0101,1100,0011,1010,0110,1111\}.$$

Nesunku patikrinti, kad ciklinio kodo apibrėžimas tenkinamas, todėl $C$ yra ciklinis kodas. □

Naudodami pereito skyriaus vektorių ir polinomų atitiktį, kodą $C$ galime nagrinėti, kaip aibės $\mathbb{F}_{q,n}[x]$ poerdvį. Tada ciklinio kodo apibrėžimo sąlyga gali būti užrašyta šitaip:

$$\text{jei } p(x)=c_0+c_1x+\dots+c_{n-1}x^{n-1}\in C,\ \text{tai } xp(x)=c_{n-1}+c_0x+\dots+c_{n-2}x^{n-1}\in C$$

(kaip matėme pereitame skyriuje, $xp(x)$ aibėje $\mathbb{F}_{q,n}[x]$ randame taip: $xp(x)=c_0x+c_1x^2+\dots+c_{n-1}x^n$ dalijame iš $x^n-1$ ir imame liekaną, t. y. tiesiog $x^n$ keičiame vienetu ir gauname $c_{n-1}+c_0x+\dots+c_{n-2}x^{n-1}$). Gauname tokį teiginį.

**4.2.3 teiginys.** *Kodas $C$ yra ciklinis tada ir tik tada, kai jis yra žiedo $\mathbb{F}_{q,n}[x]$ idealas.*

**4.2.4 pavyzdys.** 4.2.2 pavyzdžio ciklinį kodą tapatinsime su žiedo $\mathbb{F}_{q,n}[x]$ idealu

$$C=\{0,1+x^3,x+x^3,1+x,x^2+x^3,1+x^2,x+x^2,1+x+x^2+x^3\}.$$ □

### 4.2.2 Generuojantis polinomas

Kadangi visi idealai žiede $\mathbb{F}_{q,n}[x]$ yra pagrindiniai, tai egzistuoja toks polinomas $g(x)$, kad $C = \langle g(x)\rangle$, t. y. $g(x)$ yra idealo $C$ generatorius.

**4.2.5 apibrėžimas.** *Ciklinio kodo* generuojančiu polinomu *vadinsime minimalaus laipsnio nenulinį kodo polinomą su vienetiniu vyriausiuoju koeficientu.*

**4.2.6 pavyzdys.** 4.2.4 pavyzdžio ciklinio kodo $C$ generuojantis polinomas yra $1 + x$. □

**4.2.7 teorema.** *Tegu $C \neq \{0\}$ yra ciklinis kodas. Tada:*

1. *Kodas $C$ turi vienintelį generuojantį polinomą $g(x)$. Jis yra kodo $C$ generatorius.*
2. *Kodo $C$ generuojantis polinomas dalija $x^n - 1$.*
3. *Jei $g(x)$ yra kodo $C$ generuojantis polinomas,* $\deg g = r$*, tai*
$$C = \langle g(x)\rangle = \{f(x)g(x) : \deg f < n - r\}.$$
4. $\dim C = n - r$.
5. *Jei $g(x) = g_0 + g_1x + \cdots + g_{r-1}x^{r-1} + g_rx^r$, $g_r = 1$, yra kodo $C$ generuojantis polinomas, tai $g_0 \neq 0$, ir generuojanti kodo $C$ matrica yra*
$$G = \begin{pmatrix} g_0 & g_1 & g_2 & \cdots & g_r & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{r-1} & g_r & 0 & \cdots & 0 \\ 0 & 0 & g_0 & \cdots & g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & g_0 & g_1 & g_2 & \cdots & g_r \end{pmatrix}.$$

Be įrodymo.

**4.2.8 pavyzdys.** 1. Dvejetainio ciklinio kodo $C$ iš 4.2.6 pavyzdžio generuojantis polinomas $g(x) = 1 + x$ iš tiesų yra vienintelis pirmo laipsnio kodo $C$ polinomas. Nesunku patikrinti, kad tai kodo $C$ generatorius, t. y. kad $C = \langle g(x)\rangle$.

2. Polinomas $g(x)$ dalija $x^4 - 1$, nes $x^4 - 1 = (x + 1)^4$.

3. Taip pat nesunkiai įsitikiname, kad
$$\begin{aligned} C &= \{f(x)g(x) \mid \deg f < 3\} \\ &= \{0 \cdot g(x), 1 \cdot g(x), xg(x), (1 + x)g(x), x^2g(x), (1 + x^2)g(x), (x + x^2)g(x), (1 + x + x^2)g(x)\} \\ &= \{0, 1 + x, x + x^2, 1 + x^2, x^2 + x^3, 1 + x + x^2 + x^3, x + x^3, 1 + x^3\}. \end{aligned}$$

4. Iš 4.2.2 pavyzdžio matome, kad kodo $C$ dimensija yra 3. Matome, kad tai iš tikro lygu $n - r = 4 - 1 = 3$.



5. Paskutinis teoremos punktas tvirtina, kad matrica
$$G' = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$
yra kodo $C$ generuojanti matrica. Tuo galime įsitikinti, suvedę matricą $G'$ į standartinį pavidalą — matysime, kad gavome matricą $G$ iš 4.2.2 pavyzdžio. □

**4.2.9 teorema.** *Tegu $p(x) \in \mathbb{F}_q[x]$ yra polinomas su vienetiniu vyriausiuoju koeficientu. Tada $p(x)$ yra kokio nors ciklinio kodo generuojantis polinomas tada ir tik tada, kai $p(x) \mid x^n - 1$ (žymėjimas $a(x) \mid b(x)$ reiškia, kad polinomas $a(x)$ dalija polinomą $b(x)$ be liekanos).*

Be įrodymo.

Ši teorema rodo, kad egzistuoja abipusiškai vienareikšmiškas ryšys tarp polinomo $x^n - 1$ daliklių virš $\mathbb{F}_q$ aibės ir ilgio $n$ ciklinių kodų virš $\mathbb{F}_q$ aibės. Todėl visų ilgio $n$ ciklinių kodų virš $\mathbb{F}_q$ aibę galime gauti, faktorizuodami $x^n - 1$ virš $\mathbb{F}_q$ ir imdami visus jo daliklius bei jų generuotus idealus.

**4.2.10 pavyzdys.** Nustatykime, ar polinomas $g(x) = x^3 + 2x^2 + 2 \in \mathbb{F}_3[x]$ yra kurio nors ilgio 8 ciklinio kodo virš $\mathbb{F}_3$ generuojantis polinomas. Tam užtenka patikrinti, ar $g(x)$ dalija $x^8 - 1$. 4.1.3 pavyzdyje matėme, kad dalija, todėl atsakymas yra teigiamas. □

**4.2.11 pavyzdys.** Tegu $q = 3$, $n = 4$. Faktorizuokime $x^4 - 1$ virš $\mathbb{F}_3 = \{0, 1, 2\}$, kad rastume visų ilgio 4 ciklinių kodų virš $\mathbb{F}_3$ generuojančius polinomus. Pagal kvadratų skirtumo formulę iškart gauname
$$x^4 - 1 = (x^2 - 1)(x^2 + 1) = (x - 1)(x + 1)(x^2 + 1) = (x + 2)(x + 1)(x^2 + 1).$$
Lieka patikrinti, kad $f(x) = x^2 + 1$ nebesiskaido. Jei dar skaidytųsi, tai gautume $f(x) = (x - a)(x - b)$ su kažkokiom reikšmėm $a, b \in \mathbb{F}_3$, todėl gautume $f(a) = f(b) = 0$. Taigi, $f(x)$ turėtų šaknį kūne $\mathbb{F}_3$. Patikrinę gauname, kad $f(0) = 1 \neq 0$, $f(1) = 2 \neq 0$, $f(2) = 5 = 2 \neq 0$, t. y. $f(x)$ šaknų kūne $\mathbb{F}_3$ neturi, todėl ir nesiskaido.

Taigi, yra trys pirminiai $x^4 - 1$ dalikliai, todėl iš viso yra $2^3 = 8$ dalikliai:
$$\begin{aligned} g_1(x) &= 1, \\ g_2(x) &= x + 2, \\ g_3(x) &= x + 1, \\ g_4(x) &= (x + 2)(x + 1) = x^2 + 2, \\ g_5(x) &= x^2 + 1, \\ g_6(x) &= (x + 2)(x^2 + 1) = x^3 + 2x^2 + x + 2, \\ g_7(x) &= (x + 1)(x^2 + 1) = x^3 + x^2 + x + 1, \\ g_8(x) &= (x + 2)(x + 1)(x^2 + 1) = x^4 - 1. \end{aligned}$$

Pažiūrėkim, pavyzdžiui, kokį kodą $C_2$ generuoja $g_2(x)$. Jo dimensija $\dim C_2 = n - r = 4 - 1 = 3$, generuojanti matrica
$$G_2 = \begin{pmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{pmatrix} \sim \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 2 \end{array}\right),$$

todėl kontrolinė matrica $H_2 = (-2\ -2\ -2\,|\,1) = (1\,1\,1\,1)$.

Matome, kad $C_2^\perp = C_7$, nes polinomo $g_7(x)$ generuoto kodo $C_7$ generuojanti matrica pagal 4.2.7 teoremos 5 dalį yra $G_7 = (1\,1\,1\,1) = H_2$. Lygiai taip pat galime įsitikinti, kad $C_3^\perp = C_6$, $C_4^\perp = C_5$, $C_1^\perp = C_8$. Kitame skyriuje įsitikinsime, kad iš tikro ciklinio kodo dualus kodas irgi yra ciklinis kodas, taigi, jo generuojantis polinomas irgi yra vienas iš polinomo $x^n - 1$ daliklių.

Susipažinkime dar su kodu $C_4$. Jo dimensija $\dim C_4 = n - r = 4 - 2 = 2$, generuojanti matrica

$$G_4 = \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \end{pmatrix} \sim \left(\begin{array}{cc|cc} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \end{array}\right),$$

todėl kontrolinė matrica

$$H_4 = \left(\begin{array}{cc|cc} -2 & 0 & 1 & 0 \\ 0 & -2 & 0 & 1 \end{array}\right) = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Cikliniai kodai $C_1$ ir $C_8$, generuoti polinomų $g_1$ ir $g_8$, nėra įdomūs. Kodo $C_1$ dimensija $\dim C_1 = n - r = 4 - 0 = 4$, t. y. $C_1 = \mathbb{F}_3^4$. Tai $[4, 4, 1]$ kodas, t. y. koduodami mes nepridedame jokios papildomos informacijos, todėl dekoduodami, aišku, nei aptikti, o tuo labiau ištaisyti klaidų mes negalėsime. Jo generuojanti matrica $G_1$ yra $4 \times 4$ vienetinė matrica. Pagal 2.4.4 teoremą $\dim C_1^\perp = 0$, t. y. dualus kodas $C_1^\perp$ bazės, o tuo pačiu ir generuojančios matricos, neturi visai, todėl kodas $C_1$ neturi kontrolinės matricos.

Iš tikro, tas dualus kodas — tai kodas $C_8$. Polinomas $g_8$ aibėje $\mathbb{F}_{q,n}[x]$ yra lygus 0, todėl $C_8 = \langle g_8(x)\rangle = \langle 0\rangle = \{0\} = C_1^\perp$. Generuojančios matricos $C_8$ neturi, o kontrolinė yra lygi $G_1$. □

**4.2.12 pavyzdys.** Nustatykime, kiek yra ilgio $n = 12$ ciklinių kodų virš $\mathbb{F}_3$. Aišku, jų bus tiek, kiek yra polinomo $x^{12}-1$ daliklių virš $\mathbb{F}_3$. Pasinaudoję (2.2) lygybe iš 40 psl. gauname, kad $x^{12}-1 = (x^4-1)^3$. O polinomą $x^4 - 1$ virš $\mathbb{F}_3$ jau išskaidėme 4.2.11 pavyzdyje. Taigi, gauname $x^{12} - 1 = (x+2)^3(x+1)^3(x^2+1)^3$. Kiekvienas $x^{12} - 1$ daliklis bus pavidalo $(x+2)^a(x+1)^b(x^2+1)^c$, kur $0 \leqslant a, b, c \leqslant 3$. Skaičiui $a$ parinkti yra 4 galimybės, skaičiams $b$ ir $c$ taip pat, tai iš viso variantų bus $4 \cdot 4 \cdot 4 = 4^3 = 64$. Taigi, yra 64 ilgio $n = 12$ cikliniai kodai virš $\mathbb{F}_3$. □

### 4.2.3 Generatoriai

**4.2.13 pastaba.** Ciklinis kodas gali turėti kelis generatorius (4.1.7 apibrėžimo prasme), o mažiausio laipsnio generatorius su vienetiniu vyriausiuoju koeficientu vadinamas generuojančiu polinomu.

**4.2.14 pavyzdys.** 4.2.6 pavyzdžio dvejetainio ciklinio kodo $C$ generatorius yra ir $f(x) = x^3 + 1$. Iš tikro, nesunku įsitikinti, kad $\langle f(x)\rangle = C$. □

Parodysime, kaip, žinant ciklinio kodo generatorių, galima nesunkiai rasti generuojantį polinomą. Tam reikia prisiminti didžiausiojo bendrojo daliklio sąvoką.

**4.2.15 apibrėžimas.** *Tegu $f_1(x), \ldots, f_s(x) \in \mathbb{F}_q[x]$ yra polinomai. Jų* didžiausiasis bendrasis daliklis*, žymimas $\operatorname{dbd}(f_1(x), \ldots, f_s(x))$, yra didžiausio laipsnio polinomas su vienetiniu vyriausiuoju koeficientu, iš kurio dalijasi visi polinomai $f_1(x), \ldots, f_s(x)$.*



**4.2.16 pavyzdys.** Rasime polinomų $f(x) = 1+x+2x^2+x^3+x^4$ ir $g(x) = x^6-1$ virš $\mathbb{F}_3$ didžiausiąjį bendrąjį daliklį $\operatorname{dbd}(f(x), g(x))$. Prisiminkime iš algebros kurso, kaip tai daryti.

*1 būdas.* Standartinė procedūra dviejų polinomų $f(x), g(x)$ didžiausiajam bendrajam dalikliui rasti yra Euklido algoritmas. Daliname $f(x)$ iš $g(x)$ su liekana:

$$f(x) = d_1(x)g(x) + r_1(x), \quad \deg r_1(x) < \deg g(x)$$

(žr. 4.1.2 teoremą). Tada daliname $g(x)$ iš liekanos $r_1(x)$:

$$g(x) = d_2(x)r_1(x) + r_2(x), \quad \deg r_2(x) < \deg r_1(x).$$

Toliau daliname $r_1(x)$ iš liekanos $r_2(x)$:

$$r_1(x) = d_3(x)r_2(x) + r_3(x), \quad \deg r_3(x) < \deg r_2(x),$$

ir t.t. Kadangi liekanų laipsniai griežtai mažėja, tai galų gale gausime liekaną, lygią nuliui. Paskutinė nelygi nuliui liekana, dar padauginta iš tokio kūno elemento, kad vyriausiasis koeficientas taptų lygus vienetui, ir bus didžiausiasis polinomų $f(x), g(x)$ bendrasis daliklis.

Mūsų atveju dalindami kampu, kaip 4.1.3 pavyzdyje, gauname

$$\begin{aligned} x^6 - 1 &= (x^2+2x+2)(x^4+x^3+2x^2+x+1) + (2x^3+2x^2+2x), \\ x^4+x^3+2x^2+x+1 &= 2x(2x^3+2x^2+2x) + (x^2+x+1), \\ 2x^3+2x^2+2x &= 2x(x^2+x+1). \end{aligned}$$

Paskutinė nelygi nuliui liekana yra $x^2 + x + 1$, jos vyriausiasis koeficientas yra vienetas, todėl $\operatorname{dbd}(f(x), g(x)) = x^2 + x + 1$.

*2 būdas.* Mažiems polinomams kartais patogiau rasti didžiausiąjį bendrąjį daliklį yra skaidant juos į neskaidžius polinomus. Pabandykime išskaidyti $f(x)$ ir $g(x)$.

Pasinaudoję (2.2) lygybe iš 40 psl. bei kvadratų skirtumo formule, gauname

$$g(x) = x^6 - 1 = (x^2-1)^3 = (x-1)^3(x+1)^3 = (x+2)^3(x+1)^3.$$

Polinomui $f(x)$ išskaidyti reikia šiek tiek išradingumo, bet nesunkiai tai galime atlikti:

$$\begin{aligned} c(x) &= x^4+x^3+2x^2+x+1 \\ &= x^4+x^3+x^2+x^2+x+1 \\ &= x^2(x^2+x+1)+(x^2+x+1) \\ &= (x^2+x+1)(x^2+1). \end{aligned}$$

Kad $x^2 + 1$ neskaidus virš $\mathbb{F}_3$, jau matėme 4.2.11 pavyzdyje. Taip pat, kaip tame pavyzdyje, pabandome išskaidyti polinomą $h(x) = x^2+x+1$. Pastebime, kad 1 yra polinomo $h(x)$ šaknis, nes $h(1) = 3 = 0$. Taigi, $h(x)$ dalijasi iš $x - 1$. Padaliję gauname $h(x) = (x-1)^2 = (x+2)^2$. Taigi, $f(x) = (x+2)^2(x^2+1)$.

Matome, kad didžiausi neskaidžių polinomų laipsniai, įeinantys ir į $f(x)$, ir į $g(x)$ skaidinius, yra $(x+2)^2 = x^2 + x + 1$, todėl $\operatorname{dbd}(f(x), g(x)) = x^2 + x + 1$. □

**4.2.17 teorema.** *Tegu $f(x) \in \mathbb{F}_q[x]$, $n \geqslant 1$. Ilgio $n$ ciklinio kodo $\langle f(x)\rangle$ virš $\mathbb{F}_q$ generuojantis polinomas yra $\operatorname{dbd}(f(x), x^n - 1)$.*

Be įrodymo.

**4.2.18 pavyzdys.** Raskime ciklinio kodo $C$ virš $\mathbb{F}_3$, kurį generuoja žodis $c = 112110$, generuojantį polinomą. Matome, kad žodžio $c$ ilgis $n = 6$, ir jį atitinka polinomas $c(x) = 1 + x + 2x^2 + x^3 + x^4$. Todėl kodo $C = \langle c(x)\rangle$ generuojantis polinomas $g(x)$ bus $\mathrm{dbd}(c(x), x^6 - 1)$. Šių dviejų polinomų didžiausiąjį bendrąjį daliklį jau radome 4.2.16 pavyzdyje: $\mathrm{dbd}(c(x), x^6 - 1) = x^2 + x + 1$. Todėl kodo $C = \langle c(x)\rangle$ generuojantis polinomas yra $g(x) = x^2 + x + 1$. □

4.2.17 teoremą galima apibendrinti, kai turime ne vieną polinomą $f(x)$, o visą jų aibę $S = \{f_1(x), \ldots, f_s(x)\} \subset \mathbb{F}_q[x]$. Žymėkime $\langle S\rangle$ polinomų aibės $S$ generuotą žiedo $\mathbb{F}_{q,n}[x]$ idealą:

$$\langle S\rangle = \{f_1(x)a_1(x) + \cdots + f_s(x)a_s(x) \mid a_i(x) \in \mathbb{F}_{q,n}[x]\ \forall i\},$$

čia polinomai dauginami žiede $\mathbb{F}_{q,n}[x]$.

**4.2.19 teorema.** *Tegu $S = \{f_1(x), \ldots, f_s(x)\} \subset \mathbb{F}_q[x]$ yra polinomų virš $\mathbb{F}_q$ aibė, $n \geqslant 1$. Ilgio $n$ ciklinio kodo $\langle S\rangle$ virš $\mathbb{F}_q$ generuojantis polinomas yra* $\mathrm{dbd}(f_1(x), f_2(x), \ldots, f_s(x), x^n - 1)$.

Be įrodymo.

**4.2.20 pavyzdys.** Tegu $S = \{0101, 1111\}$ yra aibė dvejetainių vektorių. Raskime ciklinio kodo $\langle S\rangle$ generuojantį polinomą. Vektorių aibę $S$ sutapatinsime su atitinkamų polinomų aibe, kurią irgi žymėsime $S$: $S = \{x + x^3, 1 + x + x^2 + x^3\}$. Pažymėkime $f_1(x) = x + x^3$, $f_2(x) = 1 + x + x^2 + x^3$, $f_3(x) = x^4 - 1$. Vektorių ilgis $n = 4$. Taigi, ciklinio kodo $\langle S\rangle$ generuojantis polinomas bus $g(x) = \mathrm{dbd}(f_1, f_2, f_3)$.

Vėlgi didžiausiąjį bendrąjį daliklį galime skaičiuoti dviem būdais. Pirmas būdas — naudoti Euklido algoritmą dviejų polinomų didžiausiajam bendrajam dalikliui rasti. Galima įrodyti, kad

$$\mathrm{dbd}(f_1, \ldots, f_s) = \mathrm{dbd}(\mathrm{dbd}(f_1, \ldots, f_{s-1}), f_s).$$

Taigi, norėdami rasti $g(x) = \mathrm{dbd}(f_1, f_2, f_3)$, iš pradžių randame

$$g_1(x) = \mathrm{dbd}(f_1, f_2),$$

o paskui

$$g(x) = \mathrm{dbd}(g_1, f_3) = \mathrm{dbd}(f_1, f_2, f_3).$$

Na, bet mes gal rinkimės šiuo atveju paprastesnį sprendimą. Kadangi polinomai nedideli, lengvai juos išskaidysime neskaidžiais dauginamaisiais ir taip rasime didžiausiąjį bendrąjį daliklį:

$$\begin{aligned} f_1(x) &= x + x^3 = x(1 + x)^2, \\ f_2(x) &= 1 + x + x^2 + x^3 = 1 + x + x^2(1 + x) = (1 + x)(1 + x^2) = (1 + x)^3, \\ f_3(x) &= x^4 - 1 = (x^2 - 1)^2 = (1 + x)^4. \end{aligned}$$

Matome, kad $g(x) = \mathrm{dbd}(f_1, f_2, f_3) = (1 + x)^2 = 1 + x^2$. □



### 4.2.4 Kontrolinis polinomas

Kadangi ciklinio $[n, k]$ kodo $C$ virš $\mathbb{F}_q$ generuojantis polinomas $g(x)$ dalija $x^n - 1$, tai egzistuoja toks polinomas $h(x) \in \mathbb{F}_{q,n}[x]$, kad $x^n - 1 = g(x)h(x)$. Polinomas $h(x)$ vadinamas kodo C *kontroliniu polinomu*. Kadangi $k = \dim C = n - \deg g$, tai $\deg h = n - \deg g = n - (n - k) = k$.

**4.2.21 apibrėžimas.** *Polinomo $p(x) = p_0 + p_1x + \cdots + p_tx^t$, $p_t \neq 0$,* atvirkštiniu polinomu *vadiname polinomą $p^*(x) = x^t p(1/x) = p_0x^t + p_1x^{t-1} + \cdots + p_t$.*

**4.2.22 pavyzdys.** Jei $p(x) = 3 + x + 2x^3 + 3x^4 + 4x^5$ virš $\mathbb{F}_5$, tai $p^*(x) = 3x^5 + x^4 + 2x^2 + 3x + 4$.□

**4.2.23 teorema.** *Tarkime, $h(x) = h_0 + h_1x + \cdots + h_kx^k$ yra ciklinio kodo $C[n, k]$ virš $\mathbb{F}_q$ kontrolinis polinomas. Tada:*

1. *$C = \{p(x) \in \mathbb{F}_{q,n}[x] : p(x)h(x) = 0\}$ (dauginama aibėje $\mathbb{F}_{q,n}[x]$)*
2. *Kodo $C$ dualus kodas $C^\perp$ yra ciklinis kodas.*
3. *Kodo $C^\perp$ generuojantis polinomas yra $g^\perp(x) = h_0^{-1}h^*(x)$.*

Be įrodymo.

**4.2.24 pastaba.** 4.2.23 teoremos 3 punkte polinomą $h^*(x)$ dauginame iš $h_0^{-1}$ tam, kad vyriausiasis koeficientas taptų lygus vienetui.

**4.2.25 pastaba.** Ciklinio kodo kontrolinis polinomas $h(x)$ yra tiesinio kodo kontrolinės matricos $H$ ekvivalentas. Iš tiesų, abu jie leidžia patikrinti, ar duotas polinomas ar vektorius priklauso kodui: $Hy^T = 0 \Leftrightarrow y \in C$ ir $h(x)y(x) = 0 \Leftrightarrow y(x) \in C$. Bet jie skiriasi tuo, kad kontrolinė matrica tuo pačiu yra ir dualaus kodo generuojanti matrica, o kontrolinis polinomas bendru atveju nėra dualaus kodo generuojantis polinomas. Bet, kaip rodo teoremos 3 punktas, visgi tarp kontrolinio polinomo ir dualaus kodo generuojančio polinomo yra glaudus ryšys. Kontrolinio polinomo generuotas kodas $\langle h(x)\rangle$ yra ekvivalentus dualiam kodui $C^\perp$, nes juos gauname vieną iš kito sukeitę koordinates vietomis: pirmą su paskutine, antrą su priešpaskutine ir t.t.

**4.2.26 pavyzdžiai.** Raskime ilgio $n$ ciklinio kodo $C$ virš $\mathbb{F}_q$, kurio generuojantis polinomas yra $g(x)$, dualaus kodo generuojantį polinomą $g^\perp(x)$.

1. Tegu $n = 7$, $q = 2$, $g(x) = 1 + x + x^2 + x^4$.

   Visų pirma dalindami kampu kaip 4.1.3 pavyzdyje randame kontrolinį polinomą $h(x) = (x^7 - 1)/g(x) = x^3 + x + 1$. Tada $g^\perp(x) = h_0^{-1}h^*(x) = 1 \cdot (1 + x^2 + x^3) = 1 + x^2 + x^3$.

2. Tegu $n = 4$, $q = 3$, $g(x) = 1 + x^2$.

   Kontrolinį polinomą galime rasti dalindami $h(x) = (x^4 - 1)/g(x)$. Galime tiesiog padalinti kampu, o galime ir pasinaudoti tuo, kad $x^4 - 1$ jau išskaidėme pirminiais dauginamaisiais virš baigtinio kūno $\mathbb{F}_3$ 4.2.10 pavyzdyje:

   $$x^4 - 1 = (x + 2)(x + 1)(x^2 + 1).$$

   Todėl

   $$h(x) = (x^4 - 1)/g(x) = (x^4 - 1)/(x^2 + 1) = (x + 2)(x + 1) = x^2 + 2.$$

   Tada

   $$g^\perp(x) = h_0^{-1}h^*(x) = 2^{-1}(2x^2 + 1) = 2(2x^2 + 1) = x^2 + 2.$$ □

# Literatūra


[1] K. Bulota, P. Survila. *Algebra ir skaičių teorija*, Antrasis pataisytas ir papildytas leidimas, 1 dalis. Vilnius, Mokslas, 1989.

[2] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[3] V.Stakėnas. *Informacijos kodavimas*. Vilnius, Vilniaus universiteto leidykla, 1996.