19. poglavje

# Teorija kodiranja

- Uvod
- Enostavnejše kode za odpravljanje napak
- Glavni mejniki teorije kodiranja
- Singletonova meja
- Linearne kode
- Odkodiranje linearnih kod

Slovenski uvod:

Sandi Klavžar, O teoriji kodiranja, linearnih kodah in slikah z Marsa, *OMF* **45** (1998), 97-106.

in pa R. Jaminik, Elementi teorije informacije, ...





Na začetku so bili računalniški programi dovolj enostavni, tako da so tehnične napake (ponavadi je odpovedala elektronka) hitro postale očitne.

Z razvojem strojne opreme so postajali programi vse obsežnejši in bolj zapleteni, s tem pa je postalo upanje, da bi lahko hitro opazili majhne napake, ki spremenijo delovanje naprave, zanemarljivo in zato tudi resna skrb.

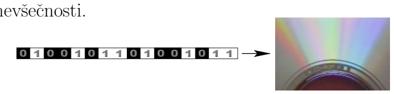
Možnost, da se nam izmuzne kakšna napaka, je vse večja tudi zato, ker so elektronska vezja iz dneva v dan manjša, računalniki pa vse hitrejši.





Tudi če je možnost napake ena sama milijardinka (npr. industrijski standard za trde diske je ena napaka na 10 milijard bitov), se bo 2GHz računalnik, zmotil približno $2\times/s$.

Glede na količino podatkov, ki jih obdelujemo dandanes, je to pravšnji recept za vsakodnevne nevšečnosti.








V času informacijske tehnologije
(zgoščenke, GSM telefoni, bančne kartice, inte
se vsi dobro zavedamo pomena hitrega in nata
prenosa, obdelovanja in hranjenja informacij.

Še tako **popolne naprave** delajo napak
pa lahko hitro spremenijo sicer izredno
programsko in strojno opremo v ničvredno
**nevarno orodje**.

Dolgo časa so se ljudje trudili izdelati računa
pomnilnike, ki bodo naredili oziroma vseboval
da malo napak (cene izdelkov pa so se višale).





Potem pa so se domislili, da bi raje računalnike same naučili iskati in odpravljati napake. Raziskovalci so našli odgovor v **kodah za odpravljanje napak**.

**Koda** je skupina simbolov, ki predstavlja informacijo. Kode obstajajo že tisočletja. To so npr.

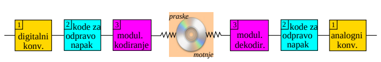
- hieroglifi,
- grška abeceda,
- rimske številke ali pa
- genetska koda za sestavljanje ribonukleinskih kislin.

Nastale so za različne potrebe:
za zapis govora ali glasbe, Morsejeva abeceda
za prenos informacij, za shranjevanje podatkov itd.





**Kode za popravljanje napak**
(angl. *error correcting codes*)
nam omogočajo, da popravljamo naključne napake, ki se pojavijo ob motnjah pri prenosu oziroma hranjenju (binarnih) podatkov.




**Claude Shannon** je postavil teoretične osnove teorije informacij in zanesljivega prenosa digitalnih podatkov kmalu po koncu druge svetovne vojne.





Za povečanje zanesljivosti prenosa in obdelave informacij smo dolgo časa uporabljali **kontrolne bite** (angl. parity-check bits), kot npr. pri številki bančnega čeka, ki pa so služili le za odkrivanje napak.

**Richard Hamming** je leta 1948 izumil metodo za *popravljanje* ene napake in *odkrivanja* dveh napak.

Ko je vnašal v računalnik programe s pomočjo luknjača kartic in mu je nato računalnik večkrat zavrnil paket kartic zaradi napak, se je zamislil:

"**Če zna računalnik sam odkriti napako, zakaj ne zna najti tudi njenega mesta in jo odpraviti.**"





**Enostavnejše kode za odpravljanje na**

Bistvo vseh metod za odpravljanja na
*dodajanje kontrolnih bitov*. Najenostavnejša
odpravljanje napak je zasnovana na **ponavlja**

Na primer, če pričakujemo, da pri prenosu
prišlo do več kot ene same napake, potem je
da ponovimo vsak bit 3× in pri sprejemu up
"**večinsko pravilo**"

**Primer:** 1101 zakodiramo v 111 111 000
prejmemo 111 011 000 111, popravimo sporoči
111 000 111 in ga končno še odkodiramo v 110

V splošnem lahko odpravimo
$n$ napak z $(2n+1)$-kratnim ponavljanjem
in uporabo večinskega pravila.

Toda ta metoda je preveč *potratna*.
V času, ko si želimo hitrega prenosa
čim večje količine podatkov,
je to popolnoma *nesprejemljivo*.

Namesto tega si želimo dodati

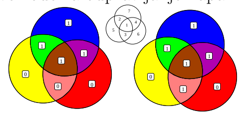
*manjše število kontrolnih bitov*,

ki bodo ravno tako ali pa še bolj učinkoviti.





Oglejmo si najpreprostejši primer
Hammingove kode za odpravljanje napak:



Zelo kratko "simfonijo" 1101 spravimo zaporedoma na rjavo (1), zeleno (2), oranžno (3) in vijoličasto (4) polje, preostala polja pa dopolnimo tako, da bo v vsakem krogu vsota števil *soda*.

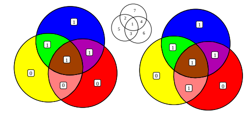
Dobimo 1101001, kjer zadnja tri mesta predstavljajo zaporedoma rumeno (5), rdeče (6) in modro (7) polje.





Naštejmo vse kodne besede, ki jih dobimo na ta način:

0000000, 0001011, 0010110, 0011101, 0100101, 0101110, 0110011, 0111000, 1000111, 1001100, 1010001, 1011010, 1100010, 1101001, 1110100, 1111111.



Recimo, da je prišlo do ene same napake in da smo prejeli vektor 1111001.

Potem bo prejemnik lahko ugotovil, da je napaka v rumenem in rdečem krogu, ne pa v modrem, kar pomeni, da je potrebno popraviti oranžno (3) polje.





Ni se težko prepričati, da je možno
način odpraviti napako na poljubnem bit
kontrolnem), pri pogoju, da je bila to edina na

S Hammingovo kodo nam je uspelo zmanjšat
kontrolnih bitov z 8 na 3, tj. dobili smo
**informacijsko stopnjo** 4/7 namesto 4/12=

Zgornjo Hammingovo kodo lahko seveda pos
Običajno to storimo z nekaj linearne algebre (





Hammingova koda odkrije, da je prišlo do napake pri prenosu tudi kadar je prišlo do dveh napak, saj ne morejo vsi trije krogi vsebovati obeh polj na katerih je prišlo do napake (če pa na dveh mestih zaznamo samo izbris, potem seveda znamo ti mesti tudi popraviti - **DN**).

Če bi tekst samo podvojili, bi dobili kodo z informacijsko stopnjo 1/2, ki pa lahko odkriva samo samostojne napake, ne more pa jih odpravljati.

V grobem lahko rečemo, da je cilj teorije kodiranja, najti **smiselen kompromis med metodo s kontrolnimi biti in metodo s ponavljanji**. Hammingova koda predstavlja prvi korak v to smer.





## Glavni mejniki teorije kodiranja

**1947-48:** začetki teorije informacij: znamenita izreka o "Source Coding" in pa "Channel Capacity" (C. Shannon)

**1949-50:** odkritje prvih kod za odpravljanje napak (M. Golay, R. Hamming).

**1959-60:** odkritje BCH-kod (R. Bose, D. Ray-Chaudhuri, A. Hochquenghem).

**1967:** Viterby algoritm za odkodiranje konvolucijskih kod.

**1993:** razvoj turbo kod (C. Berrou, A. Glavieux, P. Titimajshima).





**Teorija kodiranja** predstavlja varnostno mrežo, svojevrstno matematično zavarovanje pred muhastim materialnim svetom, v katerem živimo.

Tehnologija kod za popravljanje napak je danes tako razširjena kot zgoščenke (CD).

Omogoča nam, da poslušamo priljubljeni
Mozartov ali Madonnin CD
brez kakršnih koli motenj,
četudi nam ga mačka prav pošteno spraska.





Enako tehnologijo uporabljajo za komunikacij
vesoljske ladje in sonde, ki raziskujejo naše os

Kode za odpravljanje napak omogočajo, da pr
na Zemljo, kljub elektromagnetnim motnjam,
**kristalno jasni** posnetki oddaljenih planeto
pri tem pa za prenos porabijo

**manj energije kot hladilnikova ža**

Gre torej za

**šepetanje**,

ki mora prepotovati
več milijard km.

**Reed-Solomonove kode** doživljajo vrhunec s svojo uporabo na področju hranjenja podatkov (CD, DVD) ter prenašanja podatkov v našem osončju (te dni bo sonda **Cassini** vstopila v Saturnovo orbito in od tam pošiljala slike na Zemljo).






**Koda** je podmnožica nekega prostora z razdaljo, njeni elementi pa so **kodne besede**. **Razdalja kode** je najmanjša razdalja med različnimi kodnimi besedami.

Običajno razbijemo dano sporočilo na bloke fiksne dolžine ($n$), ki jih nato povežemo s kodnimi besedami z neko bijektivno korespondenco. V tem primeru rečemo, da gre za **bločne kode dolžine $n$**.

Najpogosteje si za prostor izberemo množico vseh $n$-teric s simboli iz neke končne množice $F$, imenovane tudi **abeceda**:

$$F^n = \{(a_0, a_1, \dots, a_{n-1}) \mid a_i \in F,\ i = 0, 1, \dots, n-1\}.$$

**Razdalja** med dvema $n$-tericama je število mest, na katerih se razlikujeta.





Pri kodi nas najbolj zanima, koliko napak lahko odpravimo, glede na to koliko kontrolnih bitov smo dodali osnovni informaciji.

**(Singletonova meja)**
*Naj bo $C$ bločna koda dolžine $n$ nad abecedo s $q$ elementi in $d$ njena razdalja. Potem velja*

$$|C| \leq q^{n-d+1}.$$

*Proof.* Naj bo $C'$ koda, ki jo konstruiramo iz kode $C$ tako, da izbrišemo skupino katerihkoli $d-1$ koordinat v vseh kodnih besedah.

Ker je razdalja kode $C$ enaka $d$, velja $|C| = |C'|$.

Dolžina kode $C'$ pa je $n-d+1$, zato ima največ $q^{n-d+1}$ kodnih besed, kar smo želeli pokazati. ■





Če so sporočila vse možne $k$-terice nad abeced
s $q$ elementi ter obstaja bijekcija med sporočil
kodnimi besedami, je $|C| = q^k$ in pravimo, da
za **$(n, k)$-kodo**.

V tem primeru se Singletonova meja prevede
v zgornjo mejo za razdaljo kode:

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





Naj bo $k$-terica $\underline{x}$ informacija, ki jo Anita zakodira v $n$-terico $\underline{y}$ ter pošlje po nekem kanalu.

Bojan prejme $n$-terico $\underline{r}$, ki ni nujno enaka $\underline{y}$, in jo odkodira po principu "**najbližjega soseda**", tj.

najprej poišče kodno besedo $\underline{y}'$, ki je najbližja $n$-terici $\underline{r}$ in nato izračuna $k$-terico $\underline{x}'$, ki se zakodira v $\underline{y}'$, v upanju, da je $\underline{y} = \underline{y}'$ in $\underline{x} = \underline{x}'$.

V tem primeru ima koda, ki odpravi $t$ napak, razdaljo $d \geq 2t+1$, saj morajo biti krogle s središčem v kodnih besedah in radijem $t$ disjunktne.





Če torej pride pri prenosu do največ $(d-1)/2$ napak, tj. $d \geq 2t+1$, se nam po principu najbližjega soseda v resnicu posreči popraviti vse napake.

Zato iz neenakosti (1) sledi, da ima taka koda vsaj $2t$ kontrolnih bitov, tj.

$$t \leq \left\lfloor \frac{n-k}{2} \right\rfloor. \qquad (2)$$

**Trditev:** *$(n, k)$-koda odpravi po principu najbližjega soseda kvečjemu $\lfloor (n-k)/2 \rfloor$ napak.*





Naj bosta $n$ in $k$ pozitivni števili, $k \leq n$.
**linearna $(n, k)$-koda $C$**
je $k$-razsežni vektorski podprostor v $\mathbb{F}^n$.

Za $k \times n$ razsežno matriko pravimo, da **generira** linearno codo $C$, če so njene vrstice baza za $C$.

Za vektorja $\underline{x}, \underline{y} \in \mathbb{F}^n$ je **Hammingova razdalja**, število kordinat, v katerih se $\underline{x}$ in $\underline{y}$ razlikujeta. Označimo jo z $d(\underline{x}, \underline{y})$,

**Razdalja** linearne $(n, k)$-kode $C$ je

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

Oznaka: **$(n, k, d)$-koda**.





## Odkodiranje v praksi

Če bi Bojan primerjal dobljeni vektor $\underline{r}$ z vsak
besedo, bi morali opraviti eksponentno število
($|C| = 2^k$) glede na $k$ (to ni polinomski algori

**Nadzorna matrika** linearne $(n, k, d)$-kode
$(n-k) \times n$-dim. binarna matrika $H$, ki gener
ortogonalni komplementa podprostora $C$.
Le-tega označimo s $C^{\perp}$ in
ga imenujemo **dualna koda** kode $C$.

Za dani vektor $\underline{r} \in \mathbb{F}^n$ naj bo $(n-k)$-terica $H\underline{r}^T$ njegov **sindrom**.

**Izrek:** *Naj bo $C$ linearna $(n,k)$-koda, ki jo generira matrika $G$, njena nadzorna matrika pa $H$. Potem za $\underline{x} \in \mathbb{F}^n$ velja*

$$\underline{x} \in C, \text{ tj. } x \text{ je kodna beseda} \iff H\underline{x}^T = 0.$$

*Če je $\underline{x} \in C$, $\underline{e} \in \mathbb{F}^n$ in $\underline{r} = \underline{x} + \underline{e}$, potem velja $H\underline{r}^T = H\underline{e}^T$ (tj. sindrom je odvisen samo od napak, ne pa tudi kodne besede).*





**Teža vektorja** $\underline{x} \in (\mathbb{F})^n$, oznaka $w(\underline{x})$, je število njegovih neničelnih koordinat, teža $(n,k)$-kode $C$ pa je

$$w(C) = \min\{w(\underline{x}) \mid \underline{x} \in C \backslash \{0\}\}.$$

**Lema:** *Če je $d$ razdalja $(n,k)$-kode $C$, potem je*

$$d = w(C).$$

**Izrek:** *Naj bo $C$ linearna $(n,k)$-koda ter $H$ njena nadzorna matrika.*

*Potem ima koda $C$ razdaljo vsaj $s$ natanko tedaj, ko je poljubnih $s-1$ stolpcev matrike $H$ linearno neodvisnih.*





## Sindromsko odkodiranje

Izračunaj $\underline{s} = H\underline{r}^T$.

Če je $\underline{s}$ ničelni vektor, odkodiraj $\underline{r}$ kot $\underline{r}$.

Sicer pa generiraj vse vektorje napak s težo 1 in njihove sindrome.

Če je za katerega od teh vektorjev $H\underline{e}^T = \underline{s}$, potem odkodiraj $\underline{r}$ kot $\underline{r} - \underline{e}$.

V nasprotnem primeru pa generiraj vse vektorje napak s težo $2, \dots, \lfloor (d-1)/2 \rfloor$ in preverjaj, ali je $H\underline{e}^T = \underline{s}$,...





Po tem postopku odkodiramo dobljeni vektor v

$$1 + \binom{n}{1} + \cdots + \binom{n}{\lfloor (d-1)/2 \rfloor}$$

korakih ali pa ugotovimo, da je prišlo do
$\lfloor (d-1)/2 \rfloor$ napak.

Medtem ko ta metoda deluje za vsako linearn
pa jo lahko za nakatere kode bistveno pospeši

V splošnem pa je odločitvena verzija tega p
NP-poln problem (kadar število napak ni om
$\lfloor (d-1)/2 \rfloor$).





Poseben primer linearnih kod, za katere obstaja hiter algoritem za odkodiranje, so **Goppa kode**.
So lahke za generiranje in imajo veliko število neekvivalentnih kod z istimi parametri.

$$n = 2^m, \quad d = 2t + 1 \ \text{ in } \ k = n - mt\ .$$

Za prakso je McEliece predlagal $m = 10$ in $t = 50$, ki nam da linearno $(1024, 524, 101)$-kodo.

Čistopis je binarna 524-terica, tajnopis pa binarna 1024-terica. Javni ključ je $(524 \times 1024)$-dim. binarna matrika.





## Opis kriptosistema McEliece

Naj bo $G$ matrika, ki generira $(n,k,d)$ Goppa kodo $C$.

Naj bo $S$ $(k \times k)$-dim. binarna matrika, ki je obrnljiva v $\mathbb{Z}_2$, $P$ $(n \times n)$-dim. permutacijska matrika in naj bo $G' = SGP$, $\mathcal{P} = (\mathbb{Z}_2)^k$, $\mathcal{C} = (\mathbb{Z}_2)^n$,

$$\mathcal{K} = \{(G, S, P, G')\}.$$

Matrika $G'$ je javna, matriki $S$ in $P$ pa tajni (privatni).

Za $K = (G, S, P, G')$ naj bo

$$e_K(\underline{x}, \underline{e}) = \underline{x}G' + \underline{e},$$

kjer je $e \in (\mathbb{Z}_2)^n$ naključni binarni vektor s težo $t$.





Bojan odšifrira tajnopis $\underline{y} \in (\mathbb{Z}_2)^n$ na naslednji način:

1. izračuna $\underline{y}_1 = \underline{y}P^{-1}$,
2. odkodira $\underline{y}_1$ tako, da najde $\underline{e}_1 = \underline{y}_1 - \underline{x}_1$, kjer je $\underline{x}_1 \in C$,
3. izračuna tak $\underline{x}_0 \in (\mathbb{Z}_2)^k$, da je $\underline{x}_0 G = \underline{x}_1$,
4. izračuna $\underline{x} = \underline{x}_0 S^{-1}$.





Za abecedo si izberimo elemente končnega ob
elementi, kjer je $q$ potenca nekega praštevila,
$\mathbb{F} = \mathrm{GF}(q)$. Če je $q$ praštevilo, je to kar praob

Potem je $\mathbb{F}^n$ z običajnim seštevanjem in mn
po komponentah vektorski prostor nad $\mathbb{F}$.

Čeprav ne bi bilo nujno, bomo obravnavo poen
in v nadaljevanju privzeli, da je dolžina kodni
enaka kar $n = q - 1$.

Multiplikativna grupa končnega obsega je $\mathbb{F}$ c

To pomeni, da obstaja v $\mathbb{F}$ **primitiven** eleme
tak element $\alpha \in \mathbb{F}$, da je $\alpha^n = 1$ in $\alpha^i \neq 1$
$i \in \{1, \dots, n-1\}$.

## Reed-Salomonove kode

Po odkritju Hammingove kode je sledilo obdobje številnih poskusov s kodami za odpravljanje napak. Ko je bila teorija kod stara 10 let sta Irving Reed in Gustave Salomon (takrat zaposlena v Lincolnovem laboratoriju na MIT) zadela v polno.

Namesto ničel in enic sta uporabila skupine bitov, ki jim tudi v računalništvu pravimo kar *besede*.





Ta lastnost je pripomogla k odpravljanju grozdnih napak, tj. napak, pri katerih se pokvari več zaporednih bitov.

Npr. šest zaporednih napak lahko pokvari največ dva bajta. Reed-Salomonova koda (na kratko R-S koda) za odpravljanje dveh napak torej predstavlja že precej dobro zaščito.

Današnje implementacije R-S kod v CD tehnologiji lahko odpravijo grozdne napake dolžine do celo 4000 bitov.





Reed in Solomon sta vpeljala RS$(n,k)$-kode s pomočjo polinomov. Za **sporočilo**

$$m = (m_0, m_1, \dots, m_{k-1}) \in \mathbb{F}^k$$

s prirejenim polinomom

$$m(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$$

izračunamo vrednosti

$$c_i = m(\alpha^i), \quad i \in \{0, \dots, n-1\}$$

in iz njih sestavimo **kodno besedo**:

$$c = (c_0, c_1, \dots, c_{n-1}).$$

Da bo odkodiranje možno, mora seveda veljati $k < n$.





V tem primeru nas dobro znana formula za pol
interpolacijo prepriča, da ni preveč pričakovat
odkodirnega algoritema za RS-kode, ki b
morebitne nepravilnosti in jih odpravil.

Bistveno vprašanje pa je, ali je tak algoritem uč

Prvi postopek za odkodiranje sta predlaga
in Solomon. Temelji na reševanju velikega
sistemov enačb.





Ko sprejmemo kodno besedo

$$c = (c_0, c_1, \dots, c_{n-1}),$$

lahko sporočilo $m = (m_0, m_1, \dots, m_{k-1})$ izračunamo iz naslednjega (predoločenega) sistema enačb

$$\begin{array}{llllll}
c_0 & = m_0 & + m_1 & + m_2 & + \cdots & + m_{k-1} \\
c_1 & = m_0 & + m_1\alpha & + m_2\alpha^2 & + \cdots & + m_{k-1}\alpha^{k-1} \\
c_2 & = m_0 & + m_1\alpha^2 & + m_2\alpha^4 & + \cdots & + m_{k-1}\alpha^{2(k-1)} \\
\vdots & & & & & \\
c_{n-1} & = m_0 & + m_1\alpha^{n-1} & + m_2\alpha^{(n-1)\cdot 2} & + \cdots & + m_{k-1}\alpha^{(n-1)(k-1)}
\end{array} \tag{3}$$





Poglejmo množico poljubnih $k$ enačb, ki ustrezajo $k$-elementni podmnožici

$$\{a_1, a_2, \dots, a_k\} \subseteq \{1, \alpha, \dots, \alpha^{n-1}\}.$$

Njihovi koeficienti tvorijo Vandermondovo matriko z determinanto

$$\begin{vmatrix} 1 & a_1 & a_1^2 & \dots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \dots & a_2^{k-1} \\ \vdots & & & & \vdots \\ 1 & a_k & a_k^2 & \dots & a_k^{k-1} \end{vmatrix} = \prod_{1 \le i < j \le k} (a_j - a_i).$$

Le-ta je v obsegu $\mathbb{F}$ različna od 0, saj je $a_i \neq a_j$ za vse $i, j \in \{1, \dots, k\}$, za katere velja $i \neq j$.

Zato ima sistem enolično rešitev v $\mathbb{F}$.





Če se pri prenosu ne bi pojavila napaka, bi lahko z izbiro poljubne $k$-elementne podmnožice obrnljivih elementov v $\mathbb{F}$ dobili sistem enačb, iz katerega bi lahko določili celotno sporočilo

$$(m_0, \dots, m_{k-1}).$$

Tako $k$-elementno podmnožico lahko izberemo na $\binom{n}{k}$ načinov.

Če pa pri prenosu nastanejo napake, nam lahko različni sistemi enačb dajo različne rešitve.

Naslednja lema nam zagotovlja, da se prava rešitev pojavi največkrat, če le število napak ni preveliko.





**LEMMA 2.** *Če pride pri prenosu ali bra*
*kodne besede* $(c_0, \dots, c_{n-1})$ *RS*$(n,k)$*-kode*
*napak, se pri reševanju podsistema k-tih e*
*iz (3) pojavi napačna rešitev (k-terica) na*

$$\binom{s+k-1}{k}\text{-krat.}$$

*Dokaz:* Enačbe sistema (3) ustrezajo $k$-ra
hiperravninam. Zaradi linearne neodvisnos
poljubnih $k$ vektorjev, ki določajo te hiperrav
se poljubnih $k$ hiperravnin seka v eni točki.

V napačni točki pa se lahko seka največ $s$ -
hiperravnin, saj je med njimi lahko največ $k-$
ki se pri prenosu niso spremenile ($k$ nesprem
enačb nam namreč že da pravo rešitev)
in največ $s$ takih, ki so se spremenile.

**IZREK 3.** *RS$(n,k)$-koda je linearna $(n,k)$-koda.*

*Dokaz:* Naj bosta $c$ in $c'$ poljubni kodni besedi RS-kode ter $m(x)$ in $m'(x)$ polinoma sporočila, katerima ustrezata ti dve kodni besedi.
Potem za $\lambda, \lambda' \in \mathbb{F}$ in $i \in \{0,1,\dots,n-1\}$ velja

$$(\lambda c + \lambda' c')_i = \lambda m(\alpha^i) + \lambda' m'(\alpha^i) = p(\alpha^i),$$

kjer je $p(x) = \lambda m(x) + \lambda' m'(x)$. Od tod sledi, da je $\lambda c + \lambda' c'$ kodna beseda, ki ustreza sporočilu $\lambda m + \lambda' m'$ in je RS-koda linearna. Kodne besede $a_i := (1, \alpha^i, \alpha^{2i}, \dots, \alpha^{(n-1)i})$ s prirejenimi polinomi $x^i$, $i \in \{0,1,\dots,k-1\}$ so linearno neodvisne, saj jih lahko zložimo v Vandermondovo matriko, katere determinanta je različna od nič, ker so števila $1, \alpha, \alpha^2, \dots, \alpha^{k-1}$ paroma različna.





Potrebno je le še preveriti, da je poljubna kodna beseda $c$, ki ustreza nekemu polinomu sporočila $m(x) = \sum_{i=0}^{k-1} m_i x^i$, linearna kombinacija le-teh:

$$c = \Big(\sum_{i=0}^{k-1} m_i (\alpha^0)^i, \sum_{i=0}^{k-1} m_i (\alpha^1)^i, \dots, \sum_{i=0}^{k-1} m_i (\alpha^{n-1})^i\Big)$$

$$= \sum_{i=0}^{k-1} m_i \big((\alpha^0)^i, (\alpha^1)^i, \dots, (\alpha^{n-1})^i\big) = \sum_{i=0}^{k-1} m_i a_i.$$

Torej je RS-koda res $k$-razsežna. ■

Sedaj pa se prepričajmo, da za RS$(n,k)$-kode v Singletonovi oceni velja enakost, tj. za dani naravni števili $n$ in $k$ RS$(n,k)$-kode odpravijo največje možno število napak.





**IZREK 4.** *RS$(n,k)$-koda odpravi $\lfloor (n-k)/2 \rfloor$ napak, njena razdalja pa je $n-k+1$.*

*Dokaz:* Privzemimo, da je pri prenosu RS-kodne besede prišlo do $s$ napak.

Po Lemi 2 dobimo pri reševanju vseh možnih podsistemov $k$-tih enačb vsako napačno rešitev

$$\text{največ} \binom{s+k-1}{k}\text{-krat, pravo pa} \binom{n-s}{k}\text{-krat.}$$

Slednje število je večje natanko tedaj, ko je

$$n - s > s + k - 1 \quad \text{oziroma} \quad s < (n-k+1)/2.$$





Ker je $s$ celo število, lahko RS-koda na ta način
poljubnih $\lfloor (n-k)/2 \rfloor$ napak.

Torej je njena razdalja vsaj $n-k+1$. ]

Iz izreka 3 sledi, da ima RS-koda $q^k$ elemento
Zaradi Singletonove meje (1) oziroma (2) pa je
enaka $n-k+1$.

Seveda je ta način za odkodiranje prepočasen,
saj zahteva reševanje $\binom{n}{k}$ sistemov enačb
velikosti $k \times k$,
kar je eksponentna časovna zahtevnost glede n