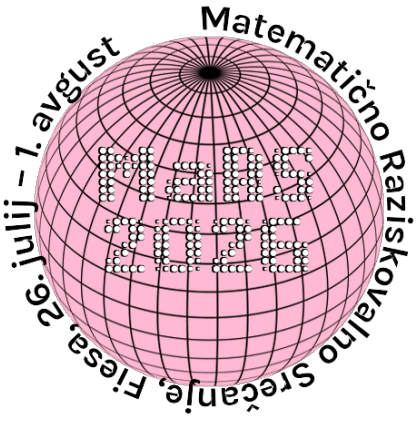
Reed-Solomonov kod

Samo Jan, Eva Šolinc
Mentor: Matija Likar



Povzetek

V članku opredelimo osnovni problem ter predstavimo ključne izreke teorije kodiranja. Definiramo Singletonovo mejo, ki opredeljuje teoretsko mejo učinkovitosti koda. Vpeljemo polinome nad končnimi polji in pokažemo temeljne izreke o njihovih ničlah. Predstavimo še kodiranje z uporabo Reed-Solomonovega koda in dokažemo, da izpolnjuje Singletonovo mejo.

1 Uvod

Dne 26. julija leta FF26 po marsovskem štetju so Zemljani želeli vzpostaviti stik z Marsovci kot najbližjimi zunajzemskimi bitji. Poslali so torej sporočilo: „Pozdravljeni!“.

Mlada Marsovca Eva in Samo sta sporočilo polna navdušenja prestregla. A navdušenju je sledilo razočaranje. Pisalo je: „gozdrarljeni!“ „Kakšni gozdarji neki?“ je zavzdihnila Eva, obupavajoča nad neumnostjo sporočila. Tedaj pa je Samo doživel preblisk: „Najbrž so želeli napisati kaj bolj smiselnega, pa se je sporočilo na poti uničilo! Pomagajva jim.“

Tako sta 1. avgusta FF26 Eva in Samo planetu Zemlji poslala sporočilo, kodirano po Reed-Solomonovem kodu. Obenem sta v medgalaktično medmrežje pripela pričujoči članek, v posebni Zemljin predalček pa položila izvirno sporočilo, da ga bodo znali razvozlati. Vse seveda v upanju, da Zemljani sploh imajo dostop do galaktičnega medmrežja.

2 Osnove teorije kodiranja

Za začetek si oglejmo nekaj osnovnih definicij, ki nam bodo pomagale opredeliti problem kodiranja in dekodiranja sporočil.

Definicija 1. ***Abeceda*** Σ *je množica znakov (npr.* $\Sigma = \{0, 1\}$*).* ***Niz*** *je* n*-terica znakov iz* Σ*, množico vseh takih* n*-teric označimo z* Σ^n*.* ***Kod*** C *je podmnožica* Σ^n*. Številu* n *pravimo tudi dolžina bloka koda.*

Za vsak kod lahko definiramo tudi **dimenzijo koda**, ki je definirana s $k = \log_q |C|$, kjer je $q = |\Sigma|$ velikost abecede. **Hitrost koda** definirano s $R = k/n$, ki nam pove gostoto informacij, ki jih pošiljamo z nekim kodom. Poznamo tudi **redudanco koda** $(n - k)$, kjer je $n \geq k$. Pojme bomo uporabljali v nadaljevanju kot metrike, ki bodo opisovale kvaliteto koda.

2.1 Pot sporočila

Začnemo s svojim sporočilom $m \in \Sigma^k$, ki ga s **kodirno funkcijo** E spremenimo v **kodirno besedo** $c \in C$. Ta gre nato skozi **kanal**, kjer se lahko zamenjajo nekateri znaki osnovnega sporočila. S tem dobimo nov niz $\mathrm{Ch}(c) \in \Sigma^n$, ki sestoji iz kodirne besede, kateri so bili nakateri znaki morda spremenjeni. Nato ta niz pošljemo skozi **dekodirno funkcijo** D, ki prejemniku rekonstruira osnovno sporočilo. Eden izmed temeljnih problemov teorije kodiranja je najti kod C ter funkciji E in D, ki omogočata rekonstrukcijo izvirnega sporočila pri čim večjem številu napak, ki se lahko pojavijo med prenosom.

$$\Sigma^k \xrightarrow{E} C \subseteq \Sigma^n \xrightarrow{\mathrm{Ch}} \Sigma^n \xrightarrow{D} \Sigma^k$$

Zgled 1. *Vzemimo $\Sigma = \{0,1\}$, $k = 3$ in $n = 9$. Sporočilo $m = (1,0,1)$ kodirna funkcija E zakodira tako, da ga trikrat ponovi.*

$$E(m) = (1,0,1,1,0,1,1,0,1) \in C \subseteq \Sigma^n$$

Recimo, da kanal spremeni znak na tretjem mestu (torej namesto 1 pošlje 0).

$$Ch(E(m)) = (1,0,0,1,0,1,1,0,1)$$

Dekodirna funkcija D za vsako od treh mest prvotnega sporočila izbere znak, ki se med tremi ponovitvami pojavi večkrat (večinsko odločanje), in tako pravilno rekonstruira

$$D(Ch(E(m))) = (1,0,1) = m.$$

Rekli smo, da se lahko med prenesom nekateri znaki v našem sporočilu spremenijo. Podrobneje si oglejmo, kako lahko te napake štejemo in kako z njimi opredelimo kvaliteto kanala.

2.2 Hammingova razdalja

Definicija 2. *Naj bosta $u, v \in \Sigma^n$. Izraz $\Delta(u,v)$ predstavlja število znakov, kjer se u in v razlikujeta, čemur pravimo Hammingova razdalja.*

Izrek 1. *Hammingova razdalja je metrika.*

Dokaz. Razdalja $\Delta(u,v)$ je po definiciji večja ali enaka nič. Hkrati je razdalja enaka nič, natanko tedaj, ko je $u = v$. Iz definicije sledi tudi simetričnost, torej $\Delta(u,v) = \Delta(v,u)$.

Velja tudi trikotniška neenakost, ki pravi,

$$\Delta(u,w) \leq \Delta(u,v) + \Delta(v,w).$$

To drži, saj če spremenimo niz u v niz v potrebujemo $\Delta(u,v)$ korakov. Če nato še nastali niz spremenimo v w pa še $\Delta(v,w)$ korakov. Torej se u in w razlikujeta kvečjemu v $\Delta(u,v) + \Delta(v,w)$ znakih. □

Dokaz trikotniške neenakosti še podkrepimo s primerom. $x = (0,0,1)$, $y = (1,0,1)$ in $z = (0,1,0)$, kjer če x prevedemo v z prek y opravimo štiri spremembe, medtem ko se niza x in z razlikujeta v dveh znakih.

2.3 Kanal s t napakami

Spomnimo se, da je kanal Ch funkcija iz Σ^n v Σ^n. Torej slika nize dolžine n v nize iste dolžine, kjer nekatere znake morebiti spremeni. Za kanal Ch pravimo, da ima t **napak**, če velja, da za vsak $v \in \Sigma^n$ velja $\Delta(v, \mathrm{Ch}(v)) \leq t$. Povedano drugače, posameznemu nizu spremeni kvečjemu t znakov. Pravimo, da kod C zaznava t napak, če obstaja dekodirna funkcija $D : \Sigma^n \to \{0,1\}$, kjer 0 pomeni, da je v kanalu prišlo do spremembe vsaj enega znaka, 1 pa, da so znaki prišli skozi brez napake. Pravimo tudi, da kod C popravi t napak, če obstaja dekodirna funkcija D, tako da za vsak $m \in \Sigma^k$ velja $D(\mathrm{Ch}(E(m))) = m$.

Zgled 2. *Za primer koda si bomo pogledali **ponovitveni kod**. S ponovitvenim kodom nekajkrat ponovimo naše osnovno sporočilo. Vzemimo $\Sigma = \{0,1\}$ in $n = 9$. Nato lahko zakodiramo sporočilo (x_1, x_2, x_3) na naslednji način.*

$$C_{3,rep}(x_1,x_2,x_3) = (x_1,x_2,x_3,x_1,x_2,x_3,x_1,x_2,x_3)$$

Našo dimenzijo koda k dobimo s preprostim računom $k = \log_2 2^3 = 3$. Zdaj lahko še izračunamo hitrost koda

$$R = \frac{k}{n} = \frac{3}{9} = \frac{1}{3},$$

kar pomeni, da naša kodirna beseda nosi trikrat več podatkov, kot bi jih bilo minimalno potrebnih za uprizoritev sporočila.

2.4 Minimalna razdalja

Kot bomo videli, nam pri izbiri koda pomaga dejstvo, da se posamezni nizi v kodu med seboj razlikujejo v čim več znakih. To raznolikost bomo izrazili z minimalno razdaljo koda.

Definicija 3. ***Minimalna razdalja*** *koda je minimalna Hammingova razdalja med poljubnima različnima kodirnima besedama.*

$$\Delta(C) = \min_{\substack{c_1, c_2 \in C \\ c_1 \neq c_2}} \Delta(c_1, c_2)$$

Na primer, bralec se lahko prepriča, da pri ponovitvenem kodu velja $\Delta(C_{3,\text{rep}}) = 3$.

Naslednji izrek je ključen, saj nam pove, da večja kot je razdalja med kodi, več napak lahko zaznamo oz. popravimo.

Izrek 2. *Naj bo C kod. Naslednje trditve so ekvivalentne.*

1. *Minimalna razdalja C je $d \geq 2$.*
2. *Kod C zazna $d - 1$ napak.*
3. *Kod C popravi $\lfloor \frac{d-1}{2} \rfloor$ napak.*

Dokaz tega izreka je na voljo v [1, Trditev 1.4.2]. Na tej točki se nam poraja vprašanje, kolikšna je največja minimalna razdalja, ki jo lahko doseže nek kod z dolžino bloka n in dimenzijo k.

Izrek 3 (Singletonova meja)**.** *Naj bo C kod z dolžino bloka n in dimenzijo k. Tedaj za njegovo minimalno razdaljo d velja*

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

Dokaz tega izreka je na voljo v [1, Izrek 4.3.1].

Če je za nek kod ta meja dosežena, pravimo, da ima kod **minimalno separabilno razdaljo**. Zdaj lahko znanje o minimalni razdalji uporabimo pri poljih. V nadaljevanju bomo videli, da je prav Reed-Solomonov kod primer koda z minimalno separabilno razdaljo.

3 Polinomi nad končnim poljem

Preden se lotimo opisa Reed-Solomonovega koda, se moramo seznaniti z nekaj definicijami in rezultati iz teorije grup in polinomov nad končnimi polji.

Definicija 4. ***Grupa*** *je par $(G, +)$, kjer je G množica, $+ \colon G \times G \to G$ pa binarna operacija, za katero velja*

- *asociativnost: za vsake $g_1, g_2, g_3 \in G$ velja $(g_1 + g_2) + g_3 = g_1 + (g_2 + g_3)$,*
- *enota: obstaja tak $e \in G$, da za vsak $g \in G$ velja $e + g = g + e = g$,*
- *inverz: za vsak $g \in G$ obstaja $g^{-1} \in G$, da velja $g + g^{-1} = e = g^{-1} + g$.*

Primer grupe je denimo množica celih števil z operacijo seštevanja, množica racionalnih števil z operacijo množenja pa ni grupa, ker število 0 nima inverza.

Definicija 5. *Če je operacija $+$ komutativna, torej velja $a + b = b + a$ za vsak par $a, b \in G$, pravimo grupi* ***Abelova grupa****.*

Definicija 6. *Polje je trojica $(F, +, \cdot)$, kjer je F množica, $+$ in $\cdot$ pa binarni operaciji, za katere velja*

- $(F, +)$ *je Abelova grupa,*
- $(F \setminus \{0\}, \cdot)$ *je Abelova grupa,*
- *distributivnost: za vsake $a, b, c \in F$ velja $a \cdot (b + c) = a \cdot b + a \cdot c$.*

Primer polja je množica racionalnih števil z operacijama seštevanja in množenja. Polja pa ne rabijo biti nujno končna. Za nas bodo prav koristna končna polja, primer katerij je naslednji. Naj bo p praštevilo. Množica $\mathbb{F}_p$ je množica $\{0, 1, 2, ..., p-1\}$ z operacijama seštevanja po $\bmod p$ in množenja po $\bmod p$ tvori polje. Dokaz, da gre res za polje, je na voljo v [1, Lema 2.1.4].

Definicija 7. ***Polinom*** *$f(x)$ nad poljem F je izraz*

$$f(x) = a_d x^d + a_{d-1} x^{d-1} + \cdots + a_1 x + a_0,$$

kjer sta $d \in \mathbb{N}_0$ in $a_i \in F$ za vsak $0 \leq i \leq d$ ter $a_d \neq 0$. Stopnja polinoma f je $\deg f = d$. Množico vseh polinomov nad poljem F označimo s $F[x]$.

Polinome lahko med seboj seštevamo in množimo. Za polinoma f in g naj velja $\deg f \leq \deg g = n$. Zapišimo ju kot $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$ in $g(x) = b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_0$. Njuna vsota znaša

$$f(x) + g(x) = (a_n + b_n) x^n + (a_{n-1} + b_{n-1}) x^{n-1} + ... + (a_1 + b_1) x + a_0 + b_0.$$

Produkt f in g je h, čigar koeficienti so

$$c_k = \sum_{0 \leq i \leq k} a_i b_{k-i}.$$

Zgled 3. *Seštevanje in množenje polinomov nad poljem bomo pokazali na primeru. Vzemimo polje $\mathbb{F}_3 = \{0, 1, 2\}$ in si izberimo polinoma $x^2 + 2x + 2$ in $x^2 + 2$, ki sta oba stopnje 2. Izračunajmo njuno vsoto in produkt.*

$$(x^2 + 2x + 2) + (x^2 + 2) = 2x^2 + 2x + 4 = 2x^2 + 2x + 1$$

$$(x^2 + 2x + 2) \cdot (x^2 + 2) = x^4 + 2x^2 + 2x^3 + 4x + 2x^2 + 4 = x^4 + 2x^3 + x^2 + x + 1$$

Spomnimo se, da smo računali po $\bmod 3$.

Opazimo lahko, da je produkt izbranih polinomov stopnje 4, vsota pa stopnje 2. V splošnem velja

$$\deg fg = \deg f + \deg g \quad \text{in} \quad \deg(f + g) \leq \max(\deg f, \deg g),$$

pri čemer enakost velja, kadar vsota vodilnih koeficientov f in g ni enaka nič.

Definicija 8. *Vrednost $a \in F$ je ničla polinoma f, če velja $f(a) = 0$.*

Bralec verjetno že ve, da je število ničel realnega polinoma omejeno z njegovo stopnjo. Podobno velja tudi za polinome nad poljubnim poljem.

Izrek 4. *Neničelni polinom $f \in F[x]$ stopnje d ima kvečjemu d različnih ničel.*

Izreka v tem članku ne bomo dokazali, lahko ga pa bralec najde v [1, Trditev 5.1.5].

Izrek 5. *Naj bodo $x_1, x_2, ..., x_k \in F$ paroma različni in naj bodo $y_1, y_2, ..., y_k$ poljubni. Obstaja enoličen polinom $f \in F[x]$ s stopnjo $\deg(f) \leq k - 1$, za katerega velja $f(x_i) = y_i$ za vsak $1 \leq i \leq k$.*

Dokaz.

Posebej bomo dokazali obstoj in enoličnost takega polinoma f.

i) Enoličnost bomo dokazali s protislovjem. Predpostavimo, da obstajata različna polinoma f in g, ki ustrezata pogojem, torej zanju velja $\deg f \leq k-1$, $\deg g \leq k-1$ in $f(x_i) = g(x_i) = y_i$ za vsak $1 \leq i \leq k$. Potem velja tudi $\deg(f-g) \leq k-1$. Posledično ima po izreku 4 polinom $f-g$ največ $k-1$ ničel.

A poglejmo še drugače. Za vsak x_j, kjer je $1 \leq j \leq k$ velja

$$(f-g)(x_j) = f(x_j) - g(x_j) = y_j - y_j = 0.$$

Polinom $f-g$ ima tako vsaj k ničel, kar je v protislovju s prejšnjo ugotovitvijo. Polinom je torej enolično določen.

ii) S konstrukcijo še dokažimo obstoj tega polinoma. Za ta namen bomo vpeljali Lagrangeev bazni polinom

$$L_i(x) = \prod_{\substack{1 \leq j \leq k \\ i \neq j}} \frac{x - x_j}{x_i - x_j}.$$

Lagrangeev bazni polinom $L_i(x_j)$ zavzame vrednost 1, če $i = j$, in 0, če $i \neq j$, o čemer naj se bralec prepriča sam.

Definirajmo polinom $f(x)$ kot

$$f(x) = \sum_{i=1}^{k} y_i L_i(x).$$

V točki x_j je tako vrednost polinoma enaka

$$f(x_j) = \sum_{i=1}^{k} y_i L_i(x_j) = y_j L_j(x_j) = y_j.$$

Dokazali smo torej, da obstaja polinom, za katerega velja $f(x_i) = y_i$ za vsak $1 \leq i \leq k$. Iz konstrukcije Lagrangeevih baznih polinomov pa tudi vidimo, da ima stopnjo največ $k-1$.

□

4 Reed-Solomonov kod

Naj bo $m = (a_0, a_1, ..., a_{k-1}) \in \mathbb{F}_p^k$ naše sporočilo, ki ga želimo poslati. Definirajmo polinom

$$f(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \cdots + a_1 x + a_0.$$

Vrednosti p in k sta med pošiljateljem in prejemnikom vnaprej dogovorjeni. Prav tako je določena vrednost $n \leq p$, ki predstavlja dolžino bloka v kodu, in različne evalvacijske točke $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_p$. Kodirna beseda je niz vrednosti polinoma $f(x)$ nad poljem $\mathbb{F}_p$ v izbranih evalvacijskih točkah. Nanančneje

$$(a_0, a_1, \ldots, a_{k-1}) \overset{E}{\mapsto} (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{n-1})).$$

Reed-Solomonov kod je množica vseh možnih kodirnih besed. V primeru, da na kanalu ne pride do napak, lahko prejemnik s poznavanjem evalvacijskih točk na podlagi izreka 5 rekonstruira polinom in s tem prvotno sporočilo.

Definicija 9. *Reed-Solomonov kod s parametroma n in k (krajše RS) je množica*

$$RS[n,k] = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) : f \in \mathbb{F}_p[x],\ \deg f < k\} \subseteq \mathbb{F}_p^n.$$

Zgled 4. *Oglejmo si še primer kodiranja sporočila po Reed-Solomonovem kodu. Sporočevalec želi poslati niz (1, 0, 1) dolžine $k = 3$. Izbere $p = 7$, torej računa v $\mathbb{F}_7$. Določi še $n = 5$ in za evalvacijske točke izbere*

$$\alpha_1 = 1, \alpha_2 = 2, \ldots, \alpha_5 = 5.$$

Pripadajoči polinom je

$$f(x) = 1 \cdot x^2 + 0 \cdot x + 1 = x^2 + 1.$$

Zdaj si ogleda vrednosti polinoma $f(x)$ v izbranih evalvacijskih točkah: $f(\alpha_1) = f(1) = 2, f(\alpha_2) = f(2) = 5, f(\alpha_3) = f(3) = 10 = 3, f(\alpha_4) = f(4) = 17 = 3, f(\alpha_5) = f(5) = 26 = 5$. Sporočevalec tako pošlje kodirno besedo (2, 5, 3, 3, 5).

Naštejmo še nekaj lastnosti Reed-Solomonovega koda.

1. Dimenzija RS je k.
2. RS je linearen kod, kar pomeni, da velja $E(m_1 + m_2) = E(m_1) + E(m_2)$ in $E(m_1\lambda) = \lambda E(m_1)$, kjer je $\lambda \in \mathbb{F}_p$ in sta m_1 in m_2 poljubni sporočili.
3. RS ima minimalno separabilno razdaljo $\Delta(RS) = n - k + 1$, zato lahko popravi $\lfloor \frac{n-k}{2} \rfloor$ napak in zazna $n - k$ napak.

V dokazu tretje lastnosti RS bomo uporabili naslednjo lemo.

Lema 1. *Naj bo C linearen kod.* ***Hammingovo težo*** $\mathrm{wt}(c)$ *definirajmo kot število neničelnih členov v kodirni besedi. Minimalna Hammingova teža neničelnih elementov v kodu C je enaka minimalni razdalji v C, torej*

$$\Delta(C) = \min_{\substack{c \in C \\ c \neq 0}} \mathrm{wt}(c).$$

Iskanje minimalne razdelja koda C smo tako prevedli na iskanje neničelne kodirne besede z minimalno Hammingovo težo. Ker mora biti kodirna beseda $(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{n-1}))$ neničelna f pa ima stopnjo kvečjemu $k - 1 < n$, mora po izreku 4 biti f neničeln polinom. Sledi torej, da je izmed $f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_{n-1})$ največ $k - 1$ enakih nič, torej je

$$\min_{\substack{c \in C \\ c \neq 0}} \mathrm{wt}(c) \geq n - (k - 1) = n - k + 1.$$

Takšen c, ki doseže mejo pa prav tako obstaja, saj lahko izberemo

$$f(x) = (x - \alpha_1)(x - \alpha_2) \cdots (x - \alpha_{n-1}).$$

Sledi, da je $\Delta(C) = n - k + 1$. Po izreku 2 vidimo, da torej Reed-Solomonov kod popravi $\lfloor \frac{n-k}{2} \rfloor$ napak.

Preostalo nam je le še opisati, kako lahko prejemnik iz kodirne besede pridobi prvotno sporočilo z napakami. Eden bolj znanih načinov, kako to storiti, je t.i. Welch-Berlekampov algoritem, ki je opisan v [3].

Literatura

[1] V. Guruswami, A. Rudra, M. Sudan, **Essential Coding Theory**, osnutek, avgust 2025. Brezplačno dostopno na `https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book`.

[2] I. S. Reed, G. Solomon, **Polynomial Codes over Certain Finite Fields**, Journal of the Society for Industrial and Applied Mathematics, 8(2):300–304, 1960. Ustanovni štiristranski članek.

[3] CS 70 Course Staff, **Note 9: Error Correcting Codes**, Discrete Mathematics and Probability Theory, UC Berkeley, pomlad 2017. Berlekamp–Welchev algoritem razloži kot sistem linearnih enačb z uvedbo polinoma napak $E(x)$. Dostopno na `https://sp17.eecs70.org/static/notes/n9.pdf`.