REED-SOLOMONOVE KODE

Aleksandar Jurišić Arjana Žitnik

6. junij 2004

Math. Subj. Class. (2000): 51E22, 94B05?, 11T71

Reed-Solomonove kode so izjemno uspešne na področju hranjenja podatkov (CD, DVD) ter prenašanja podatkov v našem osončju (sonda Cassini je po sedmih letih vstopila v Saturnovo orbito in bo naslednja štiri leta od tam pošilja slike na Zemljo). V sestavku obravnavamo Reed-Solomonove kode. Opišemo originalen pristop Reeda in Solomona ter pristop z linearnimi cikličnimi kodami. Povezava med njima je izpeljana s končno Fourierovo transformacijo. Predstavimo tudi polinomski algoritem za njihovo odkodiranje.

REED-SALOMON CODES

Reed-Solomon codes are thriving with its applications in the field of data storage (CD, DVD) and data transmition in our solar system (after seven years Cassini space probe has reached Saturn's orbit and will be sending images back to Earth for the next four years). Reed-Solomon codes are described via two different approaches: the original approach of Reed and Solomon using interpolation polynomials and the classical approach with linear cyclic codes. The connection between these two approaches is established through the finite Fourier transform. We also present a polynomial decoding algorithm.

1 Uvod

Namen transformiranja sporočila pred pošiljanjem (ali hranjenjem) informacije je lahko različen:

- dodajanje kontrolnih bitov, ki nam omogočajo odkrivanje in odpravljanje morebitnih napak po prejetju ali ponovnem branju,
- zakritje teksta pred nepooblaščeno osebo,
- kompresija teksta, da se izognemo nepotrebni informaciji.

Teorija kodiranja se ukvarja s prvo možnostjo, z drugo **kriptografija**, medtem ko ostaja zadnja običajno v domeni inženirjev. Prenosni mediji, na primer telefonske linije, omrežja in satelitske povezave, ter mediji za shranjevanje, na primer optični in magnetni diski, trakovi, običajno še zdaleč niso popolni. Če želimo zanesljive prenose in hranjenje v podatkovnih bazah, potem so kode za odpravljanje napak nepogrešljive. Eden izmed zgodnjih uspehov teorije kodiranja je kvaliteta slik, ki jo pošiljajo sateliti. Če ne bi uporabljali kod za odpravljanje napak, pošiljke ne bi vsebovale slik, na Zemljo bi prišel le naključen šum!

Vesoljska sonda Cassini je na svoji poti k Saturnu posnela Jupiter. Saturn je dosegla julija 2004, glej `http://www.ssd.rl.ac.uk/news/cassini/mission/cas.html` in od tam pošilja slike na Zemljo. Sonda je prepotovala že milijardo kilometrov, v prihodnjih štirih letih pa bo 76-krat obkrožila Saturn in izvedla 52 preletov v bližino sedmih izmed 31 znanih Saturnovih lun. Je največja medplanetarna sonda doslej, uporablja pa tudi Reed-Solomonove kode (na kratko RS-kode) za prenos podatkov na Zemljo. Cassini je skupen projekt NASA ter evropske in italijanske vesoljske agencije, ki vključuje 4.300 ljudi in vreden 3.3 milijarde dolarjev. Na dan lahko pošlje na Zemljo do 4GB podatkov.

Drugi zelo pomemben uspeh RS-kod pa so zgoščenke (CD in DVD). Njihovo matematično plat smo opisali že v Preseku [7].

V tem sestavku bomo torej spoznali kode, ki sta jih v 60. letih prejšnjega stoletja vpeljala Irving Reed in Gustave Solomon [9]. Takrat sta bila zaposlena v enem izmed Lincolnovih laboratorijev na slovitem MIT. Za svoje kode sta uporabila več kot 100 let staro teorijo končnih obsegov francoskega matematika Evarista Galoisa. Reed-Solomonov članek je predlagal zelo eleganten način procesiranja podatkov, vendar pa se nihče ni zavedal pravega pomena, npr. ali je tak način res praktičen (in tedaj verjetno sploh ni bil). Kar nekaj časa je bilo potrebno, da je tehnologija ujela korak s teorijo in omogočila učinkovite implementacije teh kod (v 60. letih prejšnjega stoletja ni bilo hitre digitalne elektronike, vsaj glede na današnje standarde). Prvi korak v to smer pa je predstavljala ugotovitev, da so Reed-Solomonove kode samo poseben primer širšega razreda cikličnih BCH-kod. Danes so RS-kode primer učinkovitih in popularnih kod na številnih področjih, kot so

- naprave za skladiščenje/hranjenje podatkov
  (trdi diski, CD, DVD ...) in njihovo branje (predvajalniki, digitalna televizija ...),
- brezžične komunikacije (mobilni telefoni, mikrovalovne povezave ...),
- satelitske komunikacije (Voyager, Mariner, Mars Lander, Cassini ...),
- modemi za širokopasovne povezave (ADSL, xDSL,...).

Prednost RS-kod je v tem, da znajo z enako lahkoto popraviti simbol z eno samo bitno napako kakor tudi simbol, pri katerem so napačni vsi biti. Zato so RS-kode posebej primerne za odpravljanje *grozdnih napak* (tj. napak, kjer se napačni biti držijo skupaj). To pa pomeni, da so RS-kode občutljive na enakomerno porazdeljene napake. Druge kode, kot npr. konvolucijske kode, so boljše za odpravljanje naključnih napak, zato pogosto bloke RS-kod še prej zakodiramo s konvolucijskimi kodami in na ta način omogočimo odpornost tako na grozdne kakor tudi na naključne napake.

Ta članek predstavlja kratek uvod v Reed-Solomonove kode, ki dosežejo Singletonovo mejo (glej naslednji razdelek). Po razdelku, v katerem na kratko opišemo najosnovnejše o kodah, je v tretjem razdelku opisan originalen pristop Reeda in Solomona. Prepričamo se, da gre za linearne kode. Po krajšem razdelku o končnih obsegih pokažemo v naslednjem razdelku s končno Fourierovo transformacijo, da so RS-kode ekvivalentne posebnemu razredu linearnih cikličnih kod. V 5. razdelku predstavimo še polinomski algoritem za odkodiranje RS-kod.

2 Osnove kodiranja

Koda je podmnožica nekega prostora, v katerem je definirana razdalja. Elementom kode bomo rekli tudi **kodne besede**. **Kodiranje** je prirejanje kodnih besed posameznim informacijam. Če pri prenosu/hranjenju kodne besede ni prišlo do napak, je **odkodiranje** obratno prirejanju. V nasprotnem primeru pa pri odkodiranju običajno prejeto besedo "popravimo" tako, da izberemo kodno besedo, ki je prejeti besedi najbližja. Temu rečemo princip **najbližjega soseda**. Precej raziskav na področju teorije kodiranja je usmerjeno v izboljšavo razmerja med verjetnostjo, da je ugibanje odkodirnega postopka pravilno, in kompleksnostjo kodiranja in odkodiranja.

Intuitivno si lahko predstavljamo prostor kot večnadstropno stanovanjsko hišo (blok), kodne besede pa so središča izbranih sob. Informacijo predstavimo (zakodiramo) tako, da v središča nekaterih izbranih sob postavimo žoge. Prenos informacij je potem podoben potresu, ki žoge prestavlja. Odkodirni proces pa poskuša žoge vrniti nazaj v središča. Če je potres zelo močan in žoga konča v drugi sobi, potem pride do napačnega popravka, kadar pa žoga ne zapusti sobe, vsaj na prvi pogled ni problema.

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

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

Razdalja kode je najmanjša razdalja med različnimi kodnimi besedami. V tem primeru je razdalja med dvema n-tericama število mest, na katerih se razlikujeta. Pravimo ji tudi **Hammingova razdalja**, po enem izmed pionirjev teorije kodiranja. Kadar je koda podmnožica takega prostora, rečemo, da gre za **bločne kode dolžine n**. Običajno razbijemo dano sporočilo na bloke fiksne dolžine (oznaka k, kjer je $0 < k \leq n$), ki jih nato povežemo s kodnimi besedami z neko bijektivno korespondenco. Taka korespondenca je posebno naravna v primeru, da kodo predstavlja k-razsežen linearen podprostor n-razsežnega vektorskega prostora. V tem primeru rečemo, da gre za **linearno (n, k)-kodo**.

Lahko pa bi si za prostor izbrali tudi kaj drugega, na primer graf, kjer je razdalja med dvema vozliščema dolžina najkrajše poti med njima. Za uvod v teorijo kodiranja glej Jurišić [7], Klavžar [8] ter Vanstone in Van Oorschot [11], za referenčno knjigo pa Pless et al. [5].

Če privzamemo odkodirni princip najbližjega soseda, ima koda, ki odpravi (do) t napak, razdaljo $d \geq 2t+1$, saj morajo biti krogle s središčem v kodnih besedah in radijem t disjunktne. Pri kodi nas najbolj zanima, koliko napak lahko odpravimo glede na to, koliko kontrolnih bitov smo dodali osnovni informaciji. V tej smeri nam pomaga naslednji rezultat.

Lema 2.1 (Singletonova meja [10]) *Naj bo C bločna koda dolžine n nad abecedo s q elementi in d njena razdalja. Potem je število elementov kode C kvečjemu q^{n-d+1}.*

Dokaz. Naj bo C' koda, ki jo konstruiramo iz kode C tako, da izberemo $d-1$ koordinat in jih zbrišemo v vseh kodnih besedah. Ker je razdalja kode C enaka d, imata obe kodi enako število elementov (iz različnih elementov kode C nismo mogli dobiti istega elementa kode C'). Koda C' ima dolžino $n-d+1$, zato ima (in s tem tudi koda C) največ q^{n-d+1} elementov. ∎

Če izberemo za sporočila vse možne k-terice nad abecedo s q elementi ter obstaja bijekcija med sporočili ter kodnimi besedami iz $C \subseteq F^n$, ima koda C q^k elementov in pravimo, da gre za **(n, k)-kodo**. V tem primeru se Singletonova meja prevede v zgornjo mejo za razdaljo kode:

$$d \leq n - k + 1. \tag{1}$$

Od tod in neenakosti $2t + 1 \leq d$ sledi, da ima koda vsaj $2t$ kontrolnih bitov, tj.

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

Torej lahko (n, k)-koda odpravi kvečjemu $\lfloor (n-k)/2 \rfloor$ napak.

Vpeljimo še nekaj oznak, ki jih bomo uporabljali skozi celoten članek. Za abecedo si izberimo elemente končnega obsega s q elementi, kjer je q potenca nekega praštevila, oznaka $\mathbb{F} = \mathrm{GF}(q)$ (angl. Galois Field). Če je q praštevilo, je $\mathrm{GF}(q)$ kar obseg $\mathbb{Z}_q$, v katerem računamo z ostanki po modulu q. Za razumevanje članka si lahko bralec, ki mu pojem končnega obsega ni domač, pod končnim obsegom predstavlja kar le-tega. Množica $\mathbb{F}^n$ z običajnim seštevanjem in množenjem po komponentah je vektorski prostor nad $\mathbb{F}$. Čeprav ne bi bilo nujno, bomo obravnavo poenostavili in v nadaljevanju privzeli, da je dolžina kodnih besed enaka kar $n = q - 1$. Dobro je znano, da je multiplikativna grupa končnega obsega $\mathbb{F}$ ciklična. To pomeni, da obstaja v $\mathbb{F}$ **primitiven** element α, tj. tak element $\alpha \in \mathbb{F}$, da je $\alpha^n = 1$ in $\alpha^i \neq 1$ za vsak $i \in \{1, \ldots, n-1\}$.

3 Polinomi

Reed in Solomon sta vpeljala RS(n,k)-kode s pomočjo polinomov. Za **sporočilo** $m = (m_0, m_1, \ldots, m_{k-1}) \in \mathbb{F}^k$ s prirejenim polinomom $m(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}$ izračunamo vrednosti $c_i = m(\alpha^i)$, $i \in \{0, \ldots, n-1\}$ in iz njih sestavimo **kodno besedo**:

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

Da bo odkodiranje možno, mora seveda veljati $k < n$. V tem primeru nas Lagrangeva formula za polinomsko interpolacijo prepriča, da ni preveč pričakovati obstoj odkodirnega algoritema za RS-kode, ki bi opazil morebitne nepravilnosti in jih odpravil. Bistveno vprašanje pa je, ali je tak algoritem učinkovit. Prvi postopek za odkodiranje sta predlagala Reed in Solomon. Le-ta temelji na reševanju velikega števila sistemov enačb. Ko sprejmemo kodno besedo $c = (c_0, c_1, \ldots, c_{n-1})$, lahko sporočilo $m = (m_0, m_1, \ldots, m_{k-1})$ izračunamo iz naslednjega (predoločenega) sistema enačb

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

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

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

Njihovi koeficienti tvorijo Vandermondovo matriko z determinanto

$$\begin{vmatrix} 1 & a_1 & a_1^2 & \ldots & a_1^{k-1} \\ 1 & a_2 & a_2^2 & \ldots & a_2^{k-1} \\ & \vdots & & & \vdots \\ 1 & a_k & a_k^2 & \ldots & a_k^{k-1} \end{vmatrix} = \prod_{1 \le i < j \le k} (a_j - a_i). \tag{4}$$

Le-ta je v obsegu $\mathbb{F}$ različna od 0, saj je $a_i \neq a_j$ za vse $i, j \in \{1, \ldots, k\}$, 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 $(m_0, \ldots, m_{k-1})$. 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.

Lema 3.1 *Če pride pri prenosu ali branju kodne besede $(c_0, \ldots, c_{n-1})$ RS(n,k)-kode do s napak, se pri reševanju podsistema k-tih enačb iz* (3) *pojavi napačna rešitev (k-terica) največ*

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

Dokaz. Enačbe sistema (3) ustrezajo hiperravninam v k-razsežnem prostoru. Zaradi linearne neodvisnosti poljubnih k vektorjev, ki določajo te hiperravnine, se poljubnih k hiperravnin seka v eni točki (4). V napačni točki pa se lahko seka največ $s+k-1$ hiperravnin, saj je med njimi lahko največ $k-1$ takih, ki se pri prenosu niso spremenile (k nespremenjenih enačb nam namreč že da pravo rešitev) in največ s takih, ki so se spremenile. Napačno točko (tj. k-terico) lahko dobimo torej na največ toliko načinov, kot smo želeli dokazati. ∎

Izrek 3.2 *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 $i \in \{0,1,\ldots,n-1\}$ 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 $a_i := (1, \alpha^i, \alpha^{2i}, \ldots, \alpha^{(n-1)i})$ s prirejenimi polinomi x^i, $i \in \{0,1,\ldots,k-1\}$ so linearno neodvisne, saj jih lahko zložimo v Vandermondovo matriko, katere determinanta je različna od nič, ker so števila $1, \alpha, \alpha^2, \ldots, \alpha^{k-1}$ 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:

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

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 odpravijo RS(n,k)-kode največje možno število napak.

Izrek 3.3 *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. Potem dobimo po Lemi 3.1 pri reševanju vseh možnih podsistemov k-tih enačb vsako napačno rešitev največ $\binom{s+k-1}{k}$-krat, pravo pa $\binom{n-s}{k}$-krat. Slednje število je večje natanko tedaj, ko je $n-s > s+k-1$ oziroma $s < (n-k+1)/2$. Ker je s celo število, lahko RS-koda na ta način odpravi poljubnih $\lfloor (n-k)/2 \rfloor$ napak. Torej je njena razdalja vsaj $n-k+1$. Iz izreka 3.2 sledi, da ima RS-koda q^k elementov. Zaradi Singletonove meje (1) pa je razdalja 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 na k. V nadaljevanju bomo spoznali tudi polinomske algoritme.

4 Računanje v končnih obsegih

Končne obsege je vpeljal Galois v tridesetih letih 19. stoletja. V slovenski literaturi so predstavljeni že v učbeniku *Algebra* Ivana Vidava [12], pred kratkim pa smo v *Preseku* [7] lahko brali tudi o računanju v (manjših) končnih obsegih. Zato si le na konkretnem primeru, ki ga bomo uporabljali v naslednjih razdelkih, oglejmo, kako v praksi seštevamo in množimo.

Obseg GF(2^4) lahko skonstruiramo z razširitvijo obsega $\mathbb{Z}_2$ z ničlo α nerazcepnega polinoma $p(x) = x^4 + x + 1$, glej npr. Vidav [12, VII.5]. Obseg GF(2^4) je vektorski prostor nad $\mathbb{Z}_2$ z bazo $\{1, \alpha, \alpha^2, \alpha^3\}$. Seštevanje v tem obsegu je čisto običajno seštevanje vektorjev (aritmetika koeficientov se izvaja v $\mathbb{Z}_2$). Za množenje je množica neničelnih elementov GF(2^4) ciklična grupa. Neničelne elemente tega obsega predstavimo kar kot potence elementa α. Ker je α ničla polinoma $p(x)$ in je karakteristika obsega enaka 2, velja $\alpha^4 = \alpha + 1$ in lahko potenco α^4 predstavimo s četverico (1100). Podobno iz $\alpha^5 = \alpha^2 + \alpha$ sledi, da lahko α^5 predstavimo z (0110), α^6 z (0011) in iz $\alpha^7 = \alpha^4 + \alpha^3 = \alpha^3 + \alpha + 1$ še α^7 z (1101). Če ta postopek nadaljujemo ter dodamo še element 0, ki ga predstavimo z α^∞, dobimo zvezo med eksponentno in vektorsko

predstavitvijo elementov obsega GF(2^4), glej tabelo 3.1. Množenje je pri t. i. eksponentni predstavitvi obsega seveda enostavnejše, npr.

$$\alpha^8 \cdot \alpha^{10} = \alpha^{18} = \alpha^3,$$

(pri zadnjem enačaju smo upoštevali $\alpha^{15} = 1$, saj je α generator multiplikativne grupe GF(2^4) in ima zato red 15), pri seštevanju v eksponentni predstavitvi pa uporabimo ZechLog tabelo (tabela 3.1), s katero prevedemo seštevanje na množenje, npr. elementa α^3 in α^5 seštejemo na naslednji način:

$$\alpha^3 + \alpha^5 = \alpha^3(1 + \alpha^2) = \alpha^3\alpha^{z(2)} = \alpha^3\alpha^8 = \alpha^{11}.$$

Morda velja opomniti, da je pri zelo velikih končnih obsegih (ki jih uporabljamo npr. v kriptografiji) sestavljanje ZechLog tabele, pa tudi njeno hranjenje, absolutno prezahtevna naloga.

α^i	0000	1000	0100	0010	0001	1100	0110	0011	1101	1010	0101	1110	0111	1111	1011	1001
i	∞	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$z(i)$	0	∞	4	8	14	1	10	13	9	2	7	5	12	11	6	3

Tabela 3.1: **ZechLog tabela** vektorske in eksponentne predstavitve obsega GF(2^4), ki je generiran z ničlo α nerazcepnega polinoma $p(x) = 1 + x + x^4$. Z njo lahko prevedemo seštevanje na množenje, saj iz enakosti $\alpha^k + \alpha^h = \alpha^{\min(k,h)}(1 + \alpha^{\max(k,h)-\min(k,h)})$ sledi, da je dovolj za vsak i najti tako število $z(i)$, da bo veljalo $1 + \alpha^i = \alpha^{z(i)}$.

5 Reed-Solomonove kode kot linearne ciklične kode

Linearna koda $C \subseteq \mathbb{F}^n$ je **ciklična**, če je za vsako kodno besedo $c = (c_0, c_1, \ldots, c_{n-1})$ tudi njen ciklični pomik, tj. beseda $c' = (c_{n-1}, c_0, c_1, \ldots, c_{n-2})$, kodna beseda. Zelo uporabno reprezentacijo ciklične kode dobimo, če kodne besede predstavimo s polinomi. Kodni besedi c podobno kot pri sporočilu priredimo polinom $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$. Cikličnemu pomiku potem ustreza polinom

$$c'(x) = c_{n-1} + c_0x + c_1x^2 + \cdots + c_{n-2}x^{n-1} = x \cdot c(x) - c_{n-1}(x^n - 1).$$

V kolobarju polinomov $R_n = \mathbb{F}[x]/(x^n - 1)$, kjer gledamo polinome po modulu polinoma $x^n - 1$, dobimo ciklični pomik kar z množenjem s polinomom x. Zato bomo pogosto enačili kodne besede s polinomi po modulu polinoma $x^n - 1$, tj. delali v kolobarju R_n.

Za obravnavo cikličnih kod je izdelana lepa in zanimiva teorija, katere osnove lahko najdemo v večini učbenikov iz teorije kodiranja, glej npr. Vanstone et al. [11], mi pa se omejimo le na najnujnejše.

Izrek 5.1 *Naj bosta n in k naravni števili, $n > k$, $g(x)$ moničen polinom (tj. polinom z vodilnim koeficientom 1) stopnje $n-k$, ki deli polinom $x^n - 1$. Potem je $S = \{a(x)g(x); \deg(a) < k\}$ cikličen podprostor vektorskega prostora R_n in $B = \{g(x), x\,g(x), \ldots, x^{k-1}g(x)\}$ baza podprostora S.*

Dokaz. Očitno je S podprostor v R_n. Pokažimo, da je S cikličen, tj. za polinom $p(x) := a(x)g(x) \in S$ moramo pokazati, da je tudi polinom $p_1(x) := x\,p(x) \mod (x^n - 1)$ v S. To je očitno, saj je razlika $p_1(x) - x\,p(x)$ deljiva z $x^n - 1$, ki je deljiv z $g(x)$, polinom $p(x)$ pa je tudi deljiv z $g(x)$. Zato je z $g(x)$ deljiv tudi polinom $p_1(x)$.

Sedaj pa pokažimo, da je množica B baza podprostora S. Predpostavimo, da je poljubna linearna kombinacija $\sum_{i=0}^{k-1} \lambda_i x^i g(x)$ enaka 0. Če obstaja največji indeks j, za katerega je $\lambda_j \neq 0$, potem je koeficient ob x^{n-k+j} enak λ_j, kar pomeni, da mora biti $\lambda_j = 0$. Torej je B res množica linearno neodvisnih vektorjev.

Enostavno preverimo še, da vektorji iz B napenjajo cel podprostor S. Vzamemo poljuben $p(x) \in S$, potem je $p(x) = a(x)g(x)$ za nek $a(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1}$. Torej je $p(x) = a_0g(x) + a_1xg(x) + \cdots + a_{k-1}x^{k-1}g(x)$ res linearna kombinacija polinomov iz $\{g(x), xg(x), \ldots, x^{k-1}g(x)\}$. ∎

Podmnožica S je **ideal** komutativnega kolobarja R, če je zaprta za seštevanje in je za vsak $r \in R$ in za vsak $s \in S$ tudi $rs \in S$. Izrek nam torej pove, da je vsak k-razsežni podprostor S, ki ustreza idealu/kodi v R_n, generiranemu s polinomom $g(x)$ stopnje $n-k$, ki deli $x^n - 1$, cikličen. Polinom $g(x)$ imenujemo **generatorski polinom** linearne ciklične kode S.

Kolobar R_n je **glavni kolobar**, tj. vsak njegov ideal je generiran z enim samim polinomom. Ni se težko prepričati, da ideali v R_n ustrezajo natanko cikličnim kodam v $\mathbb{F}^n$. Ustrezen polinom ciklične kode deli modul x^n-1. **Kodiranje** pri cikličnih kodah je potem kar množenje polinoma sporočila z generatorskim polinomom $g(x)$.

RS-kode lahko opišemo tudi kot posebne vrste linearne ciklične kode.

Izrek 5.2 *Naj bo $\mathbb{F}$ končen obseg s q elementi in $n := q-1$. Naj bo k tako število, da velja $1 \le k < n$ in $d := n-k+1$ ter α primitiven element v $\mathbb{F}$. Koda C_1 naj bo linearna ciklična koda z generatorskim polinomom $g(x) = (x-\alpha)(x-\alpha^2)\ldots(x-\alpha^{d-1})$, koda C_2 pa naj bo RS-koda, pri kateri sporočilu $m \in \mathbb{F}^k$ s prirejenim polinomom $m(x) = m_0 + m_1x + \cdots + m_{k-1}x^{k-1}$ priredimo kodno besedo $(m(\alpha), m(\alpha^2), \ldots, m(\alpha^n))$. Potem kodi C_1 in C_2 sestavljajo iste kodne besede.*

Velja opozoriti, da zgornji izrek ne trdi, da istemu sporočilu v obeh primerih priredimo isto kodno besedo in da izrek velja tudi, če pogoj $n = q-1$ zamenjamo s $(q-1) \mid n$.

Dokaz. Naj bo $c = (c_0, c_1, \ldots, c_{n-1})$ beseda iz kode C_1, ki pripada sporočilu m. Prirejeni polinom $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$ potem lahko zapišemo v obliki $c(x) = m(x)g(x)$. Prepričajmo se, da je beseda c tudi v kodi C_2. Torej je treba poiskati tak polinom $f(x)$ stopnje največ $k-1$, da bo $c_i = f(\alpha^i)$ za $i \in \{0, \ldots, n-1\}$. Naj bo $p(x) = p_0 + p_1x + \cdots + p_{n-1}x^{n-1}$, tako da velja

$$p_j = \frac{c(\alpha^{-j})}{n}, \quad j = 0, \ldots, n-1. \tag{5}$$

Izračunajmo najprej vrednosti $p(\alpha^i)$, $i \in \{0, \ldots, n-1\}$. Iz (5) sledi

$$p(\alpha^i) = \sum_{j=0}^{n-1} \frac{c(\alpha^{-j})}{n} \cdot (\alpha^i)^j = \frac{1}{n}\sum_{j=0}^{n-1}\Big(\sum_{h=0}^{n-1} c_h\alpha^{-jh}\Big)\alpha^{ij} = \frac{1}{n}\sum_{h=0}^{n-1} c_h \cdot \Big(\sum_{j=0}^{n-1}\alpha^{(i-h)j}\Big) = c_i.$$

Pri zadnjem enačaju smo upoštevali, da je izraz v zadnjem oklepaju enak n za $h = i$, sicer pa 0. To vidimo takole: α je primitiven element, zato je $\alpha^n = 1$ in $\alpha \neq 1$, se pravi, da je α ničla polinoma $(x^n-1)/(x-1) = 1 + x + x^2 + \cdots + x^{n-1}$; enako velja tudi za vse potence α, ki so različne od ena.

Polinom $c(x)$ je deljiv s polinomom $g(x)$, zato so $\alpha, \alpha^2, \ldots, \alpha^{d-1}$ tudi njegove ničle. Ker je $d-1 = n-k$, to pomeni, da za $j \in \{n-1, n-2, \ldots, k\}$ velja $c(\alpha^{-j}) = c(\alpha^{n-j}) = 0$ in zato tudi $p_j = 0$. Torej ima polinom $p(x)$ stopnjo največ $k-1$ in si ga lahko izberemo za iskani polinom $f(x)$. Ker je bilo m poljubno sporočilo iz $\mathbb{F}^k$, zaključimo, da velja $C_1 \subseteq C_2$. ∎

Pravkar dokazani izrek nam da alternativno definicijo RS-kod. Le-ta omogoča enostavno in hitro odkodiranje, ki ga bomo predstavili v naslednjem razdelku. Zgoraj opisana transformacija, ki preslika $c(x)$ v $f(x)$, je znana kot **(inverzna) Fourierova transformacija** v končnih obsegih in je diskreten analog Fourierove transformacije v analizi [3, Ch. 6].

6 Polinomski algoritem za odkodiranje

Naj bo $\mathbb{F}$ končen obseg s q elementi in $n := q - 1$. Naj bo k tako število, da velja $1 \leq k < n$ in $d := n - k + 1$. Naj bo α primitiven element v $\mathbb{F}$ in

$$g(x) = (x - \alpha)(x - \alpha^2) \dots (x - \alpha^{d-1}).$$

Obravnavamo odkodiranje pri RS(n, k)-kodi, generirani s polinomom $g(x)$. Naj bo $c(x) = a(x)g(x)$ poslana kodna beseda, $r(x)$ pa prejeta beseda. Lahko jo zapišemo v obliki

$$r(x) = c(x) + e(x), \tag{6}$$

kjer je $e(x)$ **polinom napake**. Če pri prenosu ni prišlo do napake, je $e(x)$ enak nič in je polinom $r(x)$ deljiv z $g(x)$. Polinom sporočila $a(x)$ dobimo iz $r(x)$ kar z deljenjem s polinomom $g(x)$. V primeru, da je prišlo do napake, pa bo odkodiranje težje. Opisali bomo metodo, ki odkodiranje prevede na reševanje posebnega sistema enačb.

Vemo, da obstajata taka polinoma $h(x)$ in $s(x)$, da je

$$r(x) = h(x) \cdot g(x) + s(x) \quad \text{in} \quad \deg(s(x)) < \deg(g(x)). \tag{7}$$

Polinom $s(x)$ imenujemo **sindrom** prejete besede $r(x)$. Ker so α, $\alpha^2, \dots, \alpha^{d-1}$ ničle polinoma $g(x)$ in zato tudi polinoma $c(x)$, velja zaradi (6) in (7) naslednja zveza:

$$r(\alpha^i) = e(\alpha^i) = s(\alpha^i) \quad \text{za } i = 1, \dots, d-1. \tag{8}$$

Predpostavimo, da pri prenosu ni prišlo do več kot $\ell \leq \lfloor (d-1)/2 \rfloor$ napak, kolikor jih koda največ lahko odpravi. Naj bodo $a_0, a_1, \dots, a_{\ell-1} \in \{0, \dots, n-1\}$ mesta v kodni besedi, na katerih je prišlo do napake. Potem lahko polinom $e(x)$ zapišemo v obliki

$$e(x) = \sum_{j=0}^{\ell-1} \lambda_j x^{a_j}.$$

Količino $s(\alpha^i)$ označimo s S_i. Eksponenti a_j v potenci α^{a_j} nam povedo položaje napak, zato števila α^{a_j} imenujemo **lokatorji napak**. Vrednosti λ_j pa so **velikosti napak**. Iz (8) dobimo sistem enačb

$$S_i = \sum_{j=0}^{\ell-1} \lambda_j (\alpha^i)^{a_j} = \sum_{j=0}^{\ell-1} \lambda_j (\alpha^{a_j})^i \qquad \text{za } i = 1, \dots, d-1 \tag{9}$$

z neznankami λ_j in α^{a_j}, $j = 0, \dots, \ell - 1$. Z uvedbo oznak $X_j = \alpha^{a_j}$, $j = 0, \dots, \ell - 1$, sistem (9) zapišemo v obliki

$$\begin{array}{lcccccccc} S_1 & = & \lambda_0 X_0 & + & \lambda_1 X_1 & + & \cdots & + & \lambda_{\ell-1} X_{\ell-1}, \\ S_2 & = & \lambda_0 X_0^2 & + & \lambda_1 X_1^2 & + & \cdots & + & \lambda_{\ell-1} X_{\ell-1}^2, \\ & \vdots & & & & & & & \\ S_{d-1} & = & \lambda_0 X_0^{d-1} & + & \lambda_1 X_1^{d-1} & + & \cdots & + & \lambda_{\ell-1} X_{\ell-1}^{d-1}. \end{array} \tag{10}$$

Ta sistem $d - 1$ enačb z 2ℓ neznankami (λ_j in X_j) se je v preteklosti pojavil pri reševanju različnih problemov, glej Barg [1]. Prvi se je verjetno z njim ukvarjal baron de Prony že okrog leta 1795 pri reševanju nekega interpolacijskega problema. Zanimivo je, da so različni avtorji predlagali precej podoben način za reševanje sistema (10), ki ga bomo opisali spodaj. Najprej poiščemo vrednosti X_j, nato pa lahko iz sistema (10) poiščemo še velikosti napak, saj je sistem enačb za λ_i, $i = 0, \dots, \ell - 1$, linearen.

Naj bo $\sigma(x) = 1 + \sigma_1 x + \sigma_2 x^2 + \cdots + \sigma_\ell x^\ell$ **polinom lokatorjev napake** oziroma bolj precizno polinom, ki ima za ničle ravno inverzne vrednosti lokatorjev napak, tj. $\prod_{i=0}^{\ell-1}(1-X_j x)$. Zato velja:

$$\lambda_j X_j^{\ell+u}\sigma(X_j^{-1}) = 0 \quad \text{za } j = 0, \ldots, \ell-1, \tag{11}$$

kjer je u naravno število manjše ali enako ℓ. Seštejmo enačbe (11), upoštevajmo še sistem (10) in dobimo

$$0 = \sum_{j=0}^{\ell-1} \lambda_j X_j^{\ell+u}\Big(1 + \sum_{i=1}^{\ell} \sigma_i X_j^{-i}\Big) = S_{u+\ell} + \sum_{i=1}^{\ell} \sigma_i \sum_{j=0}^{\ell-1} \lambda_j X_j^{\ell+u-i} = S_{u+\ell} + \sum_{i=1}^{\ell} \sigma_i S_{\ell+u-i},$$

kar je rekurzivna enačba za zaporedje $\{S_i\}$:

$$\sigma_1 S_{u+\ell-1} + \sigma_2 S_{u+\ell-2} + \cdots + \sigma_\ell S_u = -S_{u+\ell}. \tag{12}$$

Ko u teče od $1, \ldots, \ell$, dobimo sistem linearnih enačb za σ_i, $i = 1, \ldots \ell$, ki ga lahko zapišemo v matrični obliki

$$\begin{bmatrix} S_1 & S_2 & \ldots & S_\ell \\ S_2 & S_3 & \ldots & S_{\ell+1} \\ \vdots & \vdots & & \vdots \\ S_\ell & S_{\ell+1} & \ldots & S_{2\ell-1} \end{bmatrix} \begin{bmatrix} \sigma_\ell \\ \sigma_{\ell-1} \\ \vdots \\ \sigma_1 \end{bmatrix} = - \begin{bmatrix} S_{\ell+1} \\ S_{\ell+2} \\ \vdots \\ S_{2\ell} \end{bmatrix}. \tag{13}$$

Vnaprej ne poznamo ℓ, zato namesto z ℓ računamo z $\lfloor (d-1)/2 \rfloor$. Izkaže se, glej npr. [4, str. 149], da je rang matrike sistema v tem primeru enak številu napak. Ko poznamo število napak, lahko iz sistema (13) izračunamo koeficiente polinoma $\sigma(x)$. Da dobimo lokatorje napak, moramo poiskati ničle $\sigma(x)$ in njihove inverze. Ker smo v končnem obsegu, ničle lahko poiščemo tudi tako, da kar po vrsti preizkušamo elemente obsega (v praksi namreč obseg nima več kot 32 elementov).

Algoritem za odkodiranje Reed-Solomonovih kod, ki smo ga predstavili zgoraj, je bistveno hitrejši od tistega iz drugega razdelka, saj je polinomski. Rešimo le dva sistema enačb (13) in (10) velikosti $O(d \times d)$, iščemo inverze ℓ elementov, ki so lahko shranjeni tudi v tabeli, ter vrednosti polinoma $\sigma(x)$ v največ n točkah. Skupna zahtevnost algoritma je v najslabšem primeru enaka $O(n^3)$.

Z iskanjem algoritmov za dekodiranje RS-kod so se ukvarjali številni raziskovalci. Med njimi je bil tudi Elwyn Berlekamp, profesor elektrotehnike na kalifornijski univerzi v Berkeleyu, ki je konec 60-ih let prejšnjega stoletja odkril učinkovit algoritem za odkodiranje RS-kod [2]. Danes ga poznamo pod imenom Berlekamp-Masseyev algoritem in je izrednega pomena tudi v kriptografiji. Njegova predstavitev žal presega okvire tega članka. Odkodiranje porabi tipično do 10-krat več strojne opreme (npr. logike, pomnilnika, procesorjevih ciklov) kot kodiranje. Običajno se uporabljajo strojne implementacije RS-kod, vendar pa so danes zaradi občutno povečane hitrosti mikroprocesorjev možne tudi programske implementacije (npr. Texas Instruments daje na voljo brezplačen program za odkodiranje).

Primer 6.1 Oglejmo si kodiranje z RS$(15, 9)$-kodo nad obsegom GF(2^4), ki smo ga opisali v četrtem razdelku. Za primitivni element obsega pa si zopet izberemo ničlo α modulskega polinoma. Razdalja kode je enaka $d = 15 - 9 + 1 = 7$, tako da koda popravi do tri napake. Stopnja generatorskega polinoma $g(x)$ je $n - k = 15 - 9 = 6$. Z uporabo Tabele 3.1 izračunamo

$$g(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^3)(x-\alpha^4)(x-\alpha^5)(x-\alpha^6) = \alpha^6+\alpha^9 x+\alpha^6 x^2+\alpha^4 x^3+\alpha^{14} x^4+\alpha^{10} x^5+x^6.$$

Kodiranje je množenje s polinomom $g(x)$. Besedo $m = (0, 0, 1, 0, \alpha^{10}, 0, \alpha^2, 0, 0)$ zakodiramo torej kot $c(x) = m(x) \cdot g(x) = \alpha^6 x^2+\alpha^9 x^3+\alpha^{11} x^4+\alpha^{11} x^6+\alpha^{11} x^8+\alpha^9 x^9+\alpha^8 x^{10}+\alpha^{12} x^{11}+\alpha^2 x^{12}$ oziroma $c = (0, 0, \alpha^6, \alpha^9, \alpha^{11}, 0, \alpha^{11}, 0, \alpha^{11}, \alpha^9, \alpha^8, \alpha^{12}, \alpha^2, 0, 0)$.

Poglejmo sedaj še, kako poteka odkodiranje. Če je prirejeni polinom $c(x)$ kodne besede c deljiv s polinomom $g(x)$, potem je polinom sporočila $m(x)$ enak $c(x)/g(x)$. Poskusimo odkodirati še prejeto besedo r s prirejenim polinomom $r(x) = \alpha^6x^2 + \alpha^9x^3 + x^4 + x^5 + x^6 + \alpha^{10}x^7 + \alpha^3x^8 + \alpha^3x^9 + \alpha^2x^{12}$. Polinom $r(x)$ ni deljiv z $g(x)$, saj je ostanek enak $s(x) = \alpha^5 + \alpha^{10}x + \alpha x^2 + \alpha^{10}x^3 + \alpha^3x^4 + \alpha^9x^5$. Izračunamo $S_i = s(\alpha^i)$ za $i = 1, \ldots, 6$ in dobimo naslednje vrednosti

S_1	S_2	S_3	S_4	S_5	S_6
α^{12}	0	α^3	α^2	α^3	1

Sestavimo matriko iz sistema (13).

$$\begin{bmatrix} \alpha^{12} & 0 & \alpha^3 \\ 0 & \alpha^3 & \alpha^2 \\ \alpha^3 & \alpha^2 & \alpha^3 \end{bmatrix} \tag{14}$$

Matriko (14) enostavno prevedemo na zgornje-trikotno obliko. Od tretje vrstice odštejemo prvo, pomnoženo z α^6, in nato še drugo, pomnoženo z α^{14} (ker ima obseg karakteristiko 2, je odštevanje kar enako seštevanju). Dobimo matriko ranga 2, kar pomeni, da je pri prenosu kodne besede najverjetneje prišlo do dveh napak. Zato je treba rešiti sistem dveh enačb z dvema neznankama

$$\begin{bmatrix} \alpha^{12} & 0 \\ 0 & \alpha^3 \end{bmatrix} \begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = - \begin{bmatrix} \alpha^3 \\ \alpha^2 \end{bmatrix}, \tag{15}$$

ki nam da rešitev $\sigma_1 = \alpha^{14}$ in $\sigma_2 = \alpha^6$. Sedaj poznamo polinom $\sigma(x) = 1 + \alpha^{14}x + \alpha^6x^2$. Z računanjem njegovih vrednosti v vseh elementih obsega $\mathrm{GF}(2^4)$ preverimo, da sta njegovi ničli α^4 in α^5. Njuna inverza α^{11} in α^{10} nam povesta, da sta napaki pri prejeti besedi na desetem in enajstem mestu. Preostane nam le še, da izračunamo velikosti teh napak. V našem primeru bo to najenostavneje kar z reševanjem sistema (10). Le-ta je predoločen; če nima rešitve, je bila predpostavka, da je prišlo do največ treh napak, napačna. Velikosti napak izračunamo iz prvih dveh enačb

$$\begin{aligned} \alpha^{12} &= \lambda_0\alpha^{11} + \lambda_1\alpha^{10} \\ 0 &= \lambda_0(\alpha^{11})^2 + \lambda_1(\alpha^{10})^2 \end{aligned} \tag{16}$$

in z deljenjem s polinomom $g(x)$ preverimo, da smo res dobili kodno besedo. Velikosti napak sta $\lambda_0 = \alpha^{12}$ in $\lambda_1 = \alpha^{14}$. Polinom poslane kodne besede je potem $c_1(x) = \alpha^6x^2 + \alpha^9x^3 + x^4 + x^5 + x^6 + \alpha^{10}x^7 + \alpha^3x^8 + \alpha^3x^9 + \alpha^{14}x^{10} + \alpha^{12}x^{11} + \alpha^2x^{12}$. Ker velja $c_1(x) = g(x)\cdot(x^2 + \alpha^7x^4 + \alpha^2x^6)$, je polinom sporočila enak $x^2 + \alpha^7x^4 + \alpha^2x^6$, samo sporočilo pa je enako $(0, 0, 1, 0, \alpha^7, 0, \alpha^2, 0, 0)$.

Literatura

[1] A. Barg, *At the dawn of the theory of codes, Math. Intelligencer* **15** (1993), 20–26.

[2] E. R. Berlekamp, *Algebraic Coding Theory,* Aegean Park Press, 1968 (revised edition, 1984).

[3] Richard E. Blahut, *Algebraic Codes for Data Transmission,* Cambridge University Press, 2003.

[4] D.G. Hoffman, D.A. Leonard, C.C. Lindner, K.T. Phelps, C.A. Rodger in J.R. Wall, *Coding Theory: The Essentials*, Marcel Dekker, Inc., 1991.

[5] W. C. Huffman, V. S. Pless and R. A. Brualdi (uredniki), *Handbook of Coding Theory*, Vol. 1 & 2, North-Holland, 1998.

[6] A. Jurišić, *Računala nove dobe, 1 in 2. del*, *Presek* **30** (2002-03), str. 226-231 in 291–296.

[7] A. Jurišić, *Napake niso za vedno*, Presek, **30** (2002-03), 361-366.

[8] S. Klavžar, *O teoriji kodiranja, linearnih kodah in slikah z Marsa,* Obzornik mat. fiz. **45** (1998) 97–106.

[9] I. S. Reed and G. Solomon, *Polynomial codes over certain finite fields,* J. Soc. Indust. Appl. Math. **8** (1960) 300–304.

[10] R. C. Singleton, *Maximum distance q-nary codes*, IEEE Trans. Inform. Theory IT-10(2) (1964), 116-118.

[11] S. A. Vanstone and P. C. van Oorschot, *An Introduction to Error Correcting Codes with Applications*, Kluwer Academic Publishers, 1989.

[12] I. Vidav, *Algebra*, Mladinska knjiga, Ljubljana 1972.