UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

DARIO REJC

# KONVOLUCIJSKE KODE

DIPLOMSKO DELO

Ljubljana, 2005

**Povzetek**

*V tem delu se bomo ukvarjali s konvolucijskimi kodami, ki sodijo v razred kod za odpravljanje napak pri prenosu podatkov preko digitalnih komunikacijskih kanalov. Njihova najbolj odlikovana lastnost je izjemna učinkovitost pri odpravljanju naključno porazdeljenih napak. Konvolucijske kode bomo najprej definirali v ustreznem matematičnem okolju. Nato si bomo podrobneje ogledali njihove generatorske matrike in poiskali matrične lastnosti, ki so ključne za učinkovito vpeljavo strojnih in programskih kodirnih rešitev. Zaključili bomo s predstavitvijo Viterbijevega algoritma za odkodiranje konvolucijskih kod ter z izpeljavo zgornjih meja za nastop bitnih in grozdnih napak pri odkodiranju. Dodatno bomo predstavili osnove digitalnih komunikacijskih sistemov in nekatere najbolj odmevne uporabe konvolucijskih kod v praksi.*

**Ključne besede: pomični register, konvolucijska koda, generatorska matrika, Viterbijev algoritem, verjetnost grozdne napake, verjetnost bitne napake**

**Abstract**

*Povzetek v angleščini.*

**Key words:**

**Math. subj. class (2005):**

*Zahvala - besedilo*

# PROGRAM DIPLOMSKEGA DELA

Delo naj predstavi matematične osnove, potrebne za razumevanje konvolucijskih kod. Le-te so izjemno uporabne za odpravljanje naključnih napak, ki se pojavijo pri prenosu podatkov.

Glavni cilji so

(a) predstavitev konvolucijskih kod kot linearnih kod,

(b) pregled osnovnih lastnosti njihovih generatorskih matrik, ter

(c) predstavitev Viterbijevega odkodirnega algoritma z ocenami verjetnosti napak.

**Literatura:**


L. H. C. Lee,
*Convolutional coding, fundamentals and applications*,
Artech House Publishers, 1997.

V. C. Huffman in V. Pless,
*Fundamentals and Error-Correcting Codes*,
Cambridge University Press, 2003.

R. E. Blahut,
*Algebraic Codes for Data Transmission*,
Cambridge University Press, 2003.

R. Johannesson in K. Sh. Zigangirov,
*Fundamentals of Convolutional Coding*,
IEEE Press, 1998.

# Kazalo

# UVOD

*Kode za odpravljanje napak* naj bi zaščitile digitalne podatke pred napakami, ki se pojavljajo pri prenosu podatkov prek komunikacijskih kanalov ali shranjevanju na nezanesljive objekte (raŭnalniški spomin, trdi diski, ipd). To je posebej pomembno, ker smo v zadnjih nekaj letih priče velikemu porastu količin prenešenih podatkov kot tudi hitrosti pri njihovem shranjevanju in nenazadnje tudi izrednemu razvoju mikroelektronike. Od tod tudi potreba po čedalje zanesljivejših in učinkovitejših algoritmih za preprečevanje napak.

P. Elias leta 1955 predstavi osnovne teoretične koncepte konvolucijskih kod, njihova uporabnost pa pride do izraza v zadnjih dvajsetih letih, torej obdobju, v katerem se začne množična uporaba tako mobilnih telefonov na področju telekomunikacij kot tudi kompaktnih diskov na področju računalništva in glasbene industrije. Drugi pomemben mejnik v razvoju konvolucijskih kod predstavlja leto 1967, ko Viterbi objavi svoj algoritem za odkodiranje, ki ga razvije kot tehniko dokazovanja med svojim matematičnim raziskovanjem teorije kodiranja. Kmalu zatem pa Forney uporabi mrežno predstavitev konvolucijskih kod, s katero postane Viterbijev algoritem lahko razumljiv tudi za širše množice.

Konvolucijske kode danes skupaj z bločnimi kodami predstavljajo napogosteje uporabljane tehnike teorije kodiranja, ki zagotavljajo varen prenos in celovitost prenešenih ali shranjenih podatkov. Ključne prednosti konvolucijskih kod ležijo v naslednjih dejstvih:

(1) izjemna natančnost pri odpravljanju naključno porazdeljenih napak,

(2) zmožnost izrabe celotne informacije pri uporabi postopkov t.i. mehkega odločanja,

(3) nekonstantna dolžina kodnih besed,

(4) spomin kode oz. vpliv že procesiranih delov informacije na kodiranje trenutne informacije.

V praksi konvolucijske kode ponavadi uporabljamo skupaj z bločnimi Reed-Solomonovimi kodami. Na ta način združimo pozitivne lastnosti obeh vrst kod brez opaznih stranskih učinkov.

Cilj tega dela je predstaviti ključne koncepte konvolucijskih kod, ene izmed najpomembejših tehnik teorije kodiranja in enega najbolj ključnih zidakov med kodami za odpravljanje napak. Za razumevanje okolja, v katerem te kode uporabljamo, si bomo najprej ogledali nekaj osnovnih elementov digitalnih komunikacijskih sistemov. Temu bodo sledile osnovne definicije in izreki, ki jih bomo začeli s pridom

izkoriščati takoj, ko se bomo poglobili v raziskovanje naše osrednje teme. Nato se bomo seznanili s pomičnim registrom, ki je model za strojno izvedbo različnih tehnik teorije kodiranja. Opremljeni z vsem omenjenim bomo konvolucijske kode vmestili v ustrezno matematično okolje, potem pa še omenili nekaj koristnih grafičnih predstavitev, ki nam bodo v mnogočem olajšale razumevanje ključnih konceptov. Sledil bo podroben pregled v lastnosti generatorskih matrik, v katerem bomo izpeljali nekaj kriterijev za čim učinkovitejše uresničitve sicer popolnoma matematičnih objektov. Da zaokrožimo na začetku obljubljeno sliko osnov konvolucijskih kod bomo predstavili še Viterbijev algoritem za odkodiranje ter mu dodali ocene za zgornje meje različnih vrst napak, s katerimi bomo dobili občutek za učinkovitost tega izredno elegantnega postopka. Za zaključek pa bomo omenili še nekaj najodmevnejših dogodkov in področij, kjer so konvolucijske kode odigrale izjemno pomembno vlogo.

# Poglavje 1

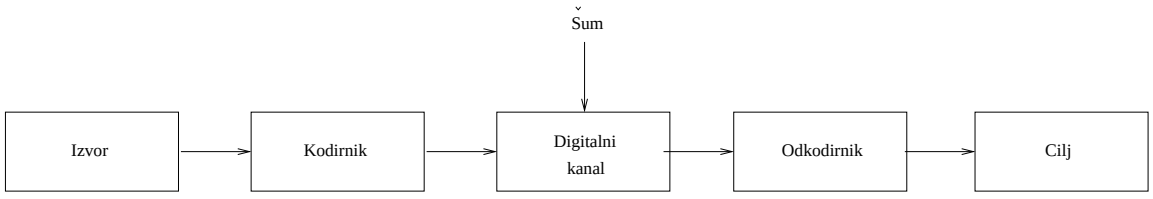
# Kodirni digitalni komunikacijski sistem

Teorija informacij temelji na dejstvu, da je vsa komunikacija v svoji osnovi digitalna, torej sestavljena iz proizvajanja, prenašanja in sprejemanja naključno izbranih binarnih cifer - *bitov*. Pri prenosu bitov pa lahko vedno pričakujemo, da bodo nekateri biti zaradi vpliva šuma in drugih dejavnikov prenešeni napačno.

## 1.1 Shannonov izrek

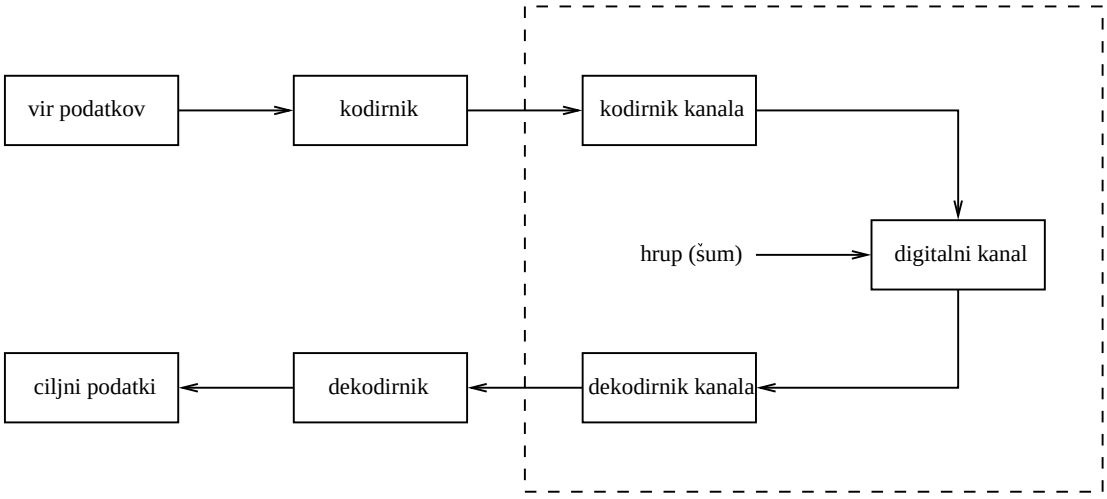
Leta 1948 je Claude E. Shannon v članku *Mathematical Theory of Communication* dokazal, da je s poljubno natančnostjo mogoče prenesti katerokoli količino podatkov po kateremkoli komunikacijskem kanalu.

Shannon je najprej pokazal, da je problem prenašanja informacije od izvora prek kanala do cilja možno ločiti (brez izgube optimalnosti) v dva podproblema in sicer na učinkovito predstavitev izvornega izhoda kot zaporedja bitov (kodiranje izvora) in prenos binarnih, slučajnih in neodvisnih bitov prek kanala (kodiranje kanala). Spodnja slika prikazuje splošni digitalni komunikacijski kanal.




Slika 1 - Splošni digitalni komunikacijski sistem

Z uporabo pravkar omenjenega Shannonovega ločitvenega principa lahko ločimo kodirnik in odkodirnik, kot je prikazano na sliki




Slika 2 - Digitalni komunikacijski sistem

To nam omogoča neodvisno implementacijo kodiranja izvora in kodiranja kanala in nam hkrati dovoljuje uporabo istih komunikacijskih kanalov pri kodiranju in prenašanju različnih informacij iz prav tako različnih izvorov.

Shannonov izrek pravi, da je možno vsaki komunikacijski kanal opisati z enim samim parametrom $C_t$, ki mu pravimo *kapaciteta kanala* in da je prek takega kanala možno poljubno natančno prenesti $R_t$ naključno izbranih bitov natanko takrat, ko je $R_t < C_t$. Količino $R_t$ imenujemo *informacijska stopnja*. Shannon je prav tako dokazal, da vrednost, ki opisuje količino šuma na dani signal, ni pomembna vse dokler velja $R_t < C_t$. Pomemben je le način, kako zakodiramo informacijske bite. Pri tem seveda ne prenašamo posameznih bitov, ampak dolga zaporedja, zakodirana tako, da ima vsak bit informacije vpliv na določeno količino bitov, prenešenih prek kanala. Ta ideja je pomenila tudi rojstvo *teorije kodiranja*.

## 1.2 Digitalni komunikacijski sistem

Komunikacijski sistem povezuje izvor podatkov z uporabnikom podatkov preko nekega kanala. Diskretni komunikacijski kanal lahko prenaša binarne simbole, simbole abecede velikosti $2^m$ in celo simbole abecede velikosti $q$, kjer $q$ ni potenca števila 2. Teorija digitalnih kombinacij nas uči, da uporaba večjih abeced (abeced z velikim številom simbolov) prinese prihranek pri energetski učinkovitosti komunikacijskega sistema.

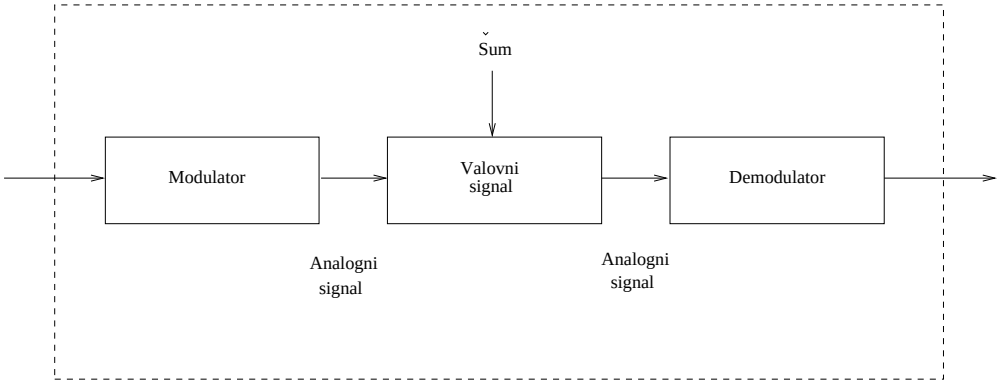
Pri sestavljanju komunikacijskega sistema je potrebno razviti naprave, ki pripravijo kodni tok podatkov kot vhod za diskretni kanal in procesirajo izhodne podatke iz diskretnega kanala tako, da uporabnik sistema lahko izlušči podatkovni tok (sporočilo). Podatke znotraj komunikacijskega sistema najpogosteje prenašamo v obliki simbolov - *bytov* - torej blokov osmih bitov. *Bit* je najmanjši gradnik vsakega digitalnega

komunikacijskega sistema in ima vrednost 0 ali 1. Včasih so simboli sestavljeni tudi iz $r$, kjer $r$ ni nujno 8. Izbira strukture simbolov znotraj sistema je razvidna uporabniku, saj le-ta lahko spremlja kodiranje in odkodiranje na vhodu in izhodu.

*Podatkovni tok* je zaporedje podatkovnih simbolov na vhodu kodirnika. *Kodni tok* je zaporedje kanalnih simbolov na izhodu kodirnika. Uporabniku se zdi, da je po kanalu poslan podatkovni tok, čeprav je tok podatkov, ki jih prenašamo po kanalu dejansko kodni tok.

Kodirnik preslika podatkovni tok v kodni tok. Tej preslikavi rečemo *koda*. Med najbolji uporabljanimi kodami srečamo dve vrsti. Prve so bločne kode, druge pa drevesne. Razlika med njimi je zasnovana na načinu uporabe spomina oziroma po tem, kako podatki, ki so že v kodirniku, vplivajo na podatke, ki se še morajo pojaviti na vhodu kodirnika.

Digitalni kanal iz slike 2 je sestavljen iz modulatorja, kanala za prenos in demodulatorja. Naslednja slika nam to razbitje tudi nazorno pokaže.




Slika 3 - Digitalni kanal

Ko pridejo podatki v komunikacijski sistem, jih najprej zakodiramo. Ponavadi je kodirnik sestavljen iz dveh ločenih komponent, kodirnika vhodnih podatkov in kanalnega kodirnika. Rezultatu kodiranja vhodnih podatkov pravimo *izvorni kodni tok*, razultatu kanalnega kodiranja pa *kanalni kodni tok*. Slednji je ponavadi daljši, saj s seboj nosi določeno količino dodatne informacije, potrebne za zaščito in pravilno rekonstrukcijo vhodnih podatkov. Kanalni kodni tok je potem poslan v modulator, ki vsak simbol tega toka prevede v simbol neke kanalne abecede. Najpogosteje rečemo, da iz kodnega toka ustvarimo signal. Ta signal nato prenesemo prek kanala. Ker je kanal izpostavljen različnim oblikam motenj, se lahko kanalni izhod razlikuje od kanalnega vhoda. Kanalni izhod je najprej spuščen skozi demodulator, ki prejeti signal ponovno prevede nazaj v zaporedje kodnih simbolov. Temu zaporedju pravimo *prejeta* ali *zaznana beseda*. Kanalni odkodirnik nato z uporabo dodatne informacije popravi napake in vrne neko kodno besedo, ki jo s pomočjo izvornega odkodirnika prevedemo v približek podatkovnega toka. Le-tega nato posredujemo uporabniku.

V prihodnjih razdelkih pa bomo pobliže spoznali nekatere ključne dele in principe delovanja digitalnega komunikacijskega sistema.

## 1.3 Elementi digitalnega komunikacijskega sistema

V tem razdelku bomo spoznali osnovne gradnike digitalnega komunikacijskega sistema.

**Vir in cilj podatkov**

Podatki v binarni obliki nastanejo v viru podatkov. Informacijske simbole ponavadi pošiljamo v obliki zaporedja, simbol za simbolom. Na ta način dobimo t.i. serijski sistem. Ključni paramater je informacijska stopnja $R_s$ vira podatkov, ki predstavlja minimalno število bitov v sekundi, ki jih potrebujemo, da lahko proizvedemo smiselen izhod iz vira. Zato poskusimo iz vira podatkov odstraniti vso odvečno informacijo, če je to le mogoče. Z načini odstranjevanja odvečne informacije iz vira podatkov se ukvarja teorija kodiranja podatkovnih virov. Te teorije se tukaj ne bomo dotikali. Zato bomo predpostavili, da so binarne cifre generirane iz vira podatkov statistično neodvisne in enako verjetno zavzamejo možne vrednosti ter nimajo redundance. Odkodirano informacijo na koncu poti pošljemo na cilj podatkov.

**Kodirnik in odkodirnik kanala**

Informacijsko zaporedje, generirano v viru podatkov, je procesirano v kodirniku kanala, ki pretvori vhodno zaporedje $k$-tih simbolov v izhodno kodno zaporedje, sestavljeno iz $n$ simbolov. Kodno zaporedje vsebuje določeno količino redundančne informacije. Informacijska stopnja prenosa, ki ga opravi kodirnik, je definirana kot razmerje $k/n$ in jo ponavadi označimo z $R_c$. Prejeto zaporedje pošljemo skozi odkodirnik, ki s pomočjo redundančne informacije poskusi popraviti napake pri prenosu, če je do teh prišlo. Odkodirnik nato predstavi približek vhodnega informacijskega zaporedja.

**Modulator, prenosna pot in demodulator**

Vsak kodirni digitalni komunikacijski sistem vsebuje modulator. Le-ta pretvori blok kodnih simbolov v ustrezen končno trajajoč signal $s(t)$ in ga tako pripravi za prenos prek kanala. Temu procesu pravimo modulacija. V splošnem rečemo, da modulator tipa $M$ bloku $\alpha$ binarnih simbolov, ki jih dobi iz kanalnega kodirnika, priredi enega

od $M$ možnih signalov, kjer je $M = 2^\alpha$ in $\alpha \geq 1$. Torej, uporabimo $\alpha$ bitov za izbiro signala, trajajočega $T$ sekund. Številu $T$ pravimo *signalni interval*.

Pri modulaciji uporabljamo več različnih tehnik ali načinov za ustvarjaje signalov. Spreminjamo lahko amplitudo, fazo ali frekvenco visokofrekvenčnih signalov (nosilcev). Če uporabljamo vhodni signal modulatorja za spreminjanje amplitude nosilca, tako dobljeno modulacijo imenujemo modulacija z zamikom amplitude s kratico ASK (angl. *amplitude-shifted keying*). Na primer, pri *M-tiški modulaciji z zamikom amplitude* je signal definiran s

$$s(t) = \begin{cases} A_i \cos(2\pi f_c t), & 0 \leq t \leq T, \\ 0, & sicer \end{cases}$$

kjer je

$$A_i = A(2i - (M-1))$$

za $i = 0, 1, \ldots, M-1$. Število $A$ je konstanta, $f_c$ pa frekvenca nosilca. Signal iz modulatorja potrebuje pasovno širino velikosti $f_c = 1/\mathrm{T}$ Hz, torej je pasovna širina obratno sorazmerna $T$.

Modulacijski sistemi s faznim zamikom so eni najbolj pogosto uporabljanih. Modulacija sloni na spreminjanju faze nosilca signala in ji zato pravimo *modulacija z zamikom faze* s kratico PSK (angl. *phase-shifted keying*).

$$s(t) = \begin{cases} A\cos(2\pi f_c t + \omega_i + \omega'), & 0 \leq t \leq T \\ 0, & sicer \end{cases}$$

kjer je

$$\omega_i = \frac{2\pi}{M} i$$

za $i = 0, 1, \ldots, M-1$. Število $A$ je spet konstanta, ki predstavlja amplitudo, $\omega'$ pa je poljuben fazni zamik, ponavadi enak 0. Tako zasnovanemu sistemu pravimo *M-tiška PSK* modulacija. Najpogosteje je v uporabi binarni ali dvojiški PSK sistem, ki ga bomo zdaj bolj podrobno opisali.

Pri binarnem PSK sistemu, modulator generira valovni signal

$$s_1(t) = \begin{cases} \sqrt{\frac{2E_s}{T}} \cos \omega t, & 0 \leq t \leq T \\ 0, & sicer \end{cases}$$

za vhod 1 in $s_0(t) = -s_1(t)$ za vhod 0 (taki modulaciji pravimo *antipodna signalizacija*). Vsak simbol traja $T$ sekund in ima energijo $E_s = ST$, kjer je $S$ moč in $\omega = 2\pi/T$. Signal, ki ga prenašamo (pošiljamo), ima obliko

$$v(t) = \sum_{i=0}^{\infty} s_{u_i}(t - iT)$$

Pri predpostavki, da na naš prenosni kanal vpliva še dodaten beli Gaussov šum $n(t)$, ki je slučajna spremenljivka z ničelnim matematičnim upanjem in kvadratom variance $N_0/2$, lahko prejeti signal zapišemo v naslednji obliki

$$r(t) = v(t) + n(t),$$

kjer je

$$E[N(T)] = 0$$

in

$$E[n(t+\tau)n(t)] = \frac{N_0}{2}\delta(\tau).$$

Na podlagi prejetega signala demodulator vrne približek $Z_i$ poslanega simbola. Približek v času $iT$ je enak

$$Z_i = \int_{(i-1)T}^{iT} r(\tau)h(iT - \tau)d\tau.$$

$Z_i$ je Gaussova slučajna spremenljivka $N(\mu, \sigma)$ z matematičnim upanjem $\mu = \pm\sqrt{E_s}$, kjer je znak odvisen od modulatorjevega vhoda (+ za 1 in − za 0), in z varianco $\sigma^2 = N_0/2$. S pomočjo pridobljenih podatkov pa se lahko trdo odločimo glede spremenljivke $Z_i$ (principa trdega in mehkega odločanja bomo opisali v petem poglavju). Na ta način dobimo najenostavnejši in najpomembnejši model kanala z binarnim vhodom in binarnim izhodom, t.i. *binarni simetrični kanal* s prehodno verjetnostjo $\varepsilon$. Ker pa je izhod kanala odvisen le od prenešenega signala v danem intervalu in ni odvisen od ostalih prenosov, rečemo da je kanal brez spomina.

## 1.4 Modeli kanalov

Za popolnejšo sliko opišimo še nekaj osnovnih modelov kanalov, po katerih prenašamo podatke.

**Diskretni kanal brez spomina (DMC)**

Pri kodiranju se ne splača uporabljati trdnih odločitev, saj lahko to pomeni izgubo informacije. Ker vsak bit informacije vpliva na nekaj kanalnih simbolov, odkodirnik lahko iz vrednosti $Z_i$ pridobi informacijo o zanesljivosti prejetega simbola (trdne

odločitve uporabljajo le predznak $Z_i$). Na podlagi tega pa demodulator pošlje analogno vrednost $Z_i$ kot svoj izhod. Temu postopku rečemo *mehko odločanje*. Z uporabo tega postopka dobimo tudi novo vrsto kanala za prenos podatkov, ki mu pravimo *diskretni kanal brez spomina*. Tukaj bomo spoznali najpomembnejša predstavnika:

(a) binarni simetrični kanal

(b) kanal z binarnim vhodom in osmiškim izhodom

Oglejmo si ju bolj podrobno.

### Binarni simetrični kanal (BSC)

Najpogostejši diskretni kanal brez spomina je *binarni simetrični kanal*. To je kanal z binarnim vhodom in binarnim izhodom, ki ga dodatno opišemo s pomočjo *verjetnosti preskoka* $\varepsilon$. Kot bomo videli, je verjetnost preskoka tesno povezana z razmerjem $E_s/N_0$. Binarno PSK shemo z binarnim simetričnim kanalom z verjetnostjo preskoka $\varepsilon$ in verjetnostjo bitne napake $P_b$ podamo z

$$\varepsilon = P_b = Q(\sqrt{\frac{2E_s}{N_0}}) = Q(\sqrt{\frac{2E_b}{N_0}}),$$

kjer je

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-\beta^2/2} d\beta$$

in je povprečna energija na signalni simbol $E_s$ enaka povprečni energiji na informacijski bit $E_b$. Do take ugotovitve pridemo iz dejstva, da je en sam informacijski bit potreben za kreacijo signalnega simbola. Spodnja slika kaže binarni simetrični kanal z verjetnostjo preskoka $p$.

Slika 4 - Binarni simetrični kanal z verjetnostjo preskoka $p$

Količina $P(x \mid y)$ pomeni pogojno verjetnost, da smo prejeli $x$, pri pogoju, da je bil poslan $y$.

### Kanal z binarnih vhodom in osmiškim izhodom (BSC)

Kanal, ki ga bomo predstavili, bo osnova za algoritem z mehkim odločanjem, ki ga bomo raziskali v petem poglavju. Imenujemo ga *kanal z binarnim vhodom in osmiškim izhodom*. Zakaj, bo razvidno iz sledečega opisa.

Naloga demodulatorja je iz prejetega simbola določiti poslani simbol. Označimo s $\boldsymbol{c} = c_1 \dots c_n$ zaporedje, ki ga prenašamo po kanalu, z $\boldsymbol{v'} = v'_1 \dots v'_n$ zaporedje, ki ga dobimo iz demodulatorja ter z $\boldsymbol{v} = v_1 \dots v_n$ zaporedje, ki predstavlja prejeto sporočilo, iz katerega bomo potem poiskušali dobiti naše originalno sporočilo. Poudarimo, da so elementi $v'_i$ realna števila, medtem, ko $c_i$ in $v_i$ lahko zavzamejo le vrednosti 0 in 1. Zdaj razdelimo realno premico v disjunktno unijo osmih intervalov z nekih sedmih vrednosti. Intervalom priredimo vrednosti $0_1$, $0_2$, $0_3$, $0_4$, $1_1$, $1_2$, $1_3$ in $1_4$. Denimo, na primer, da intervalu $(a, b]$ priredimo vrednost $0_1$, kjer je lahko $a = -\infty$ ali $b = \infty$. Z drugimi besedami, za pravi simbol $v_i$ izberemo $0_1$ takrat, ko je $v'_i \in (a, b]$. Pogojna verjetnost, da je $v'_i \in (a, b]$, pri pogoju, da je bil prenešen $c_i$, je enaka

$$P(0_1 \mid c_i) = P(a < v'_i \leq b \mid c_i) = \frac{1}{\sqrt{2\pi}\sigma} \int_a^b e^{\frac{-(v-\mu)^2}{2\sigma^2}} dv,$$

kjer je $\mu = \sqrt{E_s}$, če je $c_i = 1$ in $\mu = -\sqrt{E_s}$, če je $c_i = 0$. Torej lahko, z izbiro sedmih realnih vrednosti in z uporabo enačb, podobnih zgornji, izračunamo 16 pogojnih verjetnosti $P(v \mid c)$, kjer $v$ teče od $0_1$ do $1_4$, $c$ pa je lahko 0 ali 1. Tem verjetnostim, pri predpostavki, da na kanal vpliva beli Gaussov šum, pravimo *statistika kanala*. Izračunamo jih lahko, če poznamo energijo signala $E_s$ in dvostransko spektralno gostoto moči $N_0/2$, kjer upoštevamo, da je $\sigma^2 = N_0/2$. Ker $c_i$ lahko zavzamejo dve vrednosti in predstavljajo vhod modulatorja in ker $v_i$ lahko zavzame osem vrednosti, ki predstavljajo izhod demodulatorja, pravimo takemu kanalu *diskretni kanal brez spomina z binarnim vhodom in osmiškim izhodom*. Tak kanal prikažemo grafično kot kaže spodnja slika.

Slika 5- Diskretni kanal brez spomina z binarnim vhodom in osmiškim izhodom

Verjetnosti $P(v \mid c)$ seveda lahko napišemo tudi nad povezave, tako kot smo to naredili v primeru binarnega simetričnega kanala.

Seveda je vsota vseh verjetnosti, ki izhajajo iz 0, enaka 1 in enako velja za vsoto verjetnosti, ki izhajajo iz 1. Ponavadi oznake izberemo tako, da štiri največje verjetnosti, izhajajoče iz 0, določajo $0_1, \ldots, 0_4$ in podobno za $1_1, \ldots, 1_4$.

Med kanali, ki jih srečamo v uporabi je še kanal z medsimbolno interferenco, a njegov opis bomo tukaj izpustili. S tem končujemo opis osnovnih elementov digitalnega komunikacijskega sistema.

# Poglavje 2

# Osnove

Drugo poglavje nas bo popeljalo čez nekaj osnovnih pojmov in lastnosti, ki predstavljajo osnove teorije kodiranja in so nujno potrebne za razumevanje poglavij, ki sledijo.

## 2.1 Kode

Naj bo $V$=$\{0,1\}$ in naj bo $V^n$ množica vseh besed dolžine $n$, sestavljenih iz 0 in 1. Torej je:

$$V^n = \{b_1 b_2 \dots b_n \mid b_i \in V, i = 1, 2, \dots, n\}$$

.

V množico $V^n$ vpeljemo še operaciji seštevanja po komponentah po modulu 2 in množenja s skalarjem. Na ta način postane $(V^n, +, \cdot)$ vektorski prostor.

**Definicija 2.1.1. Dvojiška koda dolžine $n$** *je podmnožica $C$ množice $V^n$. Elemente podmnožice $C$ imenujemo* **kodne besede**.

Opozorimo na dejstvo, da lahko namesto izbrane množice $V$ izberemo poljubno končno množico simbolov (tako množico ponavadi imenujemo *abeceda*). V pričujočem delu bomo uporabljali izključno dvojiške kode, saj le-te povsem zadoščajo za vse potrebne izpeljave in so v praksi tudi največkrat uporabljane. Zaradi pravkar povedanega, bomo tudi dvojiškim kodam odslej rekli na kratko kode. Dodajmo še, da ne predstavlja vsaka podmnožica množice $V^n$ uporabne kode.

Seveda si bomo želeli povedati kaj več tudi o odnosih med kodnimi besedami. Eden osnovnih pojmov v teoriji kodiranja je t.i. *Hammingova razdalja*, ki nam dá oceno, koliko narazen sta kodni besedi. Naj bosta torej $u, v \in V^n$.

**Definicija 2.1.2. Hammingova razdalja** *je taka funkcija $d_H : V^n \times V^n \to \mathbb{N} \cup 0$, ki paru kodnih besed priredi naravno število, ki je enako številu mest (bitov), v katerih se dani besedi razlikujeta.*

**Trditev 2.1.3.** *Množica $V^n$, opremljena s Hammingovo razdaljo, je metrični prostor.*

**Dokaz.** Za dokaz zgornje trditve je dovolj pokazati, da Hammingova razdalja $d_H$ zadošča aksiomom razdalje

(1) $d_H(u, v) \geq 0$.

(2) Če je $d_H(u, v) = 0$, potem sledi $u = v$.

(3) $d_H(u, v) = d_H(v, u)$.

Prvi trije aksiomi sledijo neposredno iz definicije Hammingove razdalje.

(4) Trikotniška neenakost: $d_H(u, t) \leq d_H(u, v) + d_H(v, t)$.

Označimo z $w_H(u + v)$ razdaljo $d_H(u, v)$. Ustrezno vpeljimo oznaki $w_H(v + t)$ in $w_H(u + t)$. Količina $w_H(x)$ je enaka številu enic v besedi $x$. Pravimo ji *teža besede.*

**Definicija 2.1.4. Teža besede** *je število neničelnih komponent v tej besedi. Drugače povedano, teža besede je enaka Hammingovi razdalji te besede od ničelne besede.*

Ker za poljubni besedi $a$ in $b$ velja

$$w_H(a) + w_H(b) \geq w_H(a + b),$$

saj se istoležne enice pokrajšajo, s substitucijo $a = u + v$ in $b = v + t$ v zgornji enačbi dobimo trikotniško neenakost. (Seveda smo ves čas izvajali seštevanje po modulu 2).

Ker zadošča razdalja $d_H$ vsem potrebnim aksiomom, je $(V^n, d_H)$ metrični prostor. □

Hammingova razdalja nam pove, koliko narazen sta kodni besedi. Vpeljimo s pomočjo Hammingove razdalje tudi prvo lastnost kode $C$.

**Definicija 2.1.5. Minimalna razdalja $\delta(C)$ kode** $C$ *je najmanjša razdalja med poljubnima kodnima besedama, v znakih:*

$$\delta(C) = \min\{d_H(u, v) |\ u, v \in C\}$$

V primerih, ko bo nedvoumno, o kateri kodi govorimo, bomo uporabljali kar $\delta$ namesto $\delta(C)$.

Pri odkrivanju in popravljanju napak , ki so nastale med prenosom po kanalu se bomo držali osnovnega principa teorije kodiranja, imenovanega *princip najbližjega soseda*. Oglejmo si, kaj pravi.

*Denimo, da je $w$ beseda, ki smo jo prejeli po kanalu. Med vsemi kodnimi besedami poiščemo tisto kodno besedo $u$, za katero je $d_H(w,u) \leq d_H(w,v)$ za vse kodne besede $v$, $v \neq u$. Tedaj besedo $w$ odkodiramo v besedo $u$, ker je v tem primeru beseda $u$ najbližja glede na Hammingovo razdaljo.*

Omenjeni pristop je uporaben takrat, ko pri prenosu besed ne pride do velikega števila napak. Princip najbližjega soseda pove, da koda popravi $e$ napak takrat, ko za vsako prejeto besedo $w$ velja, da obstaja kvečjemu ena kodna beseda $u$ z lastnostjo $d_H(w,u) \leq e$. Torej, če pri prenosu besede pride do kvečjemu $e$ napak, nam princip najbližjega soseda zagotavlja, da bomo besedo pravilno odkodirali, to je, da napake pri prenosu ne vplivajo na pravilnost odkodiranja.

Število napak, ki jih znamo odpraviti z uporabo principa najbližjega soseda, je tesno povezano z minimalno razdaljo $\delta(C)$ kode $C$.

**Trditev 2.1.6.** *(1) Koda z minimalno razdaljo $\delta$ odkrije $\delta - 1$ napak.*

*(2) Koda z minimalno razdaljo $\delta$ popravi po principu najbližjega soseda $e$ napak natanko takrat, ko velja neenakost $\delta \geq 2e + 1$.*

**Dokaz.**

(1) Očitno je $\delta \geq 1$. Recimo, da je pri pošiljanju kodne besed $u$ prišlo do kvečjemu $\delta - 1$ napak. Prejeta beseda potem ni enaka nobeni izmed kodnih besed, saj je $\delta$ minimalna razdalja uporabljene kode. Iz tega lahko sklepamo, da je pri prenosu prišlo do napake.

(2) ($\Rightarrow$) Predpostavimo, da je $\delta \leq 2e$. Izberimo kodni besedi $u$ in $v$, za kateri velja $d_H(u,v) = \delta$. Tedaj lahko iz $u$ dobimo $v$ tako, da spremenimo $\delta$ bitov. Označimo z $w$ besedo, ki jo dobimo po opravljenih $\lfloor \delta/2 \rfloor$ korakih. Tedaj je $d_H(w,u) \leq e$ in $d_H(w,v) \leq e$. To je protislovje, saj bi po principu najbližjega soseda morala obstajati kvečjemu ena kodna beseda $t$, za katero bi veljalo $d_H(w,t) \leq t$.

($\Leftarrow$) Naj bo zdaj $\delta \geq 2e + 1$. Če bi beseda $w$ bila oddaljena za kvečjemu $e$ od kodnih besed $u$ in $v$ (torej napak bi bilo manj kot $e$), bi iz trikotniške neenakosti sledilo

$$d_H(u,v) \leq d_H(u,w) + d_H(w,v) \leq e + e = 2e,$$

kar je v nasprotju s predpostavko.

□

Iz pravkar dokazane trditve sledi:

**Posledica 2.1.7.** *Koda z minimalno razdaljo $\delta$ odkrije natanko $\delta - 1$ napak in popravi $\lfloor(\delta-1)/2\rfloor$ napak.* □

Izkoristimo dejstvo, da je $(V^n, d_H)$ metrični prostor in definirajmo kroglo $K_r(u)$ s polmerom $r$ okrog besede $u$:

$$K_r(u) = \{w \in V^n \mid d_H(u,w) \leq r\}.$$

S to terminologijo trditev 2.1.5(2) pravi, da
*koda popravi e napak natanko tedaj, kadar so krogle s polmerom e okrog kodnih besed paroma disjunktne.*
Ena izmed opor pri iskanju novih kod so različne meje, ki povezujejo dolžino kode, število njenih kodnih besed in njeno minimalno razdaljo. Verjetno najbolj znana med njimi je Singletonova meja.

**Izrek 2.1.8.** [**Singletonova meja**] *Naj koda $C$ dolžine $n$ vsebuje $M$ kodnih besed, Tedaj velja $M \leq 2^{n-\delta-1}$.*

**Dokaz.** Naj bo $C'$ množica $M$ besed, ki jih dobimo tako, da iz kodnih besed izbrišemo $\delta - 1$ bitov (seveda, v vsaki besedi izbrišemo bite na istih mestih, tj. koordinatah). Besede v $C'$ so torej dolge $n - (\delta - 1)$ in so paroma različne, saj so besede v $C$ bile paroma oddaljene vsaj $\delta$. Ker je besed dolžine $n - (\delta - 1)$ kvečjemu $2^{n-(\delta-1)}$, je izrek dokazan. □

## 2.2 Linearne kode

V množici $V^n$ definiramo seštevanje dveh besed kot besedo, ki jo dobimo s seštevanjem istoležnih bitov po modulu 2. S tako definiranim seštevanjem postane $(V^n, +)$ Abelova grupa. Če definiramo še množenje s skalarjema 0 in 1, postane $(V^n, +, \cdot)$ vektorski prostor nad obsegom $\mathbb{Z}_2$.

**Definicija 2.2.1.** *Koda $C \subseteq V^n$ je* **linearna**, *če je $(C, +, \cdot)$ vektorski podprostor prostora $(V^n, +, \cdot)$.*

Množenje s skalarjem iz $\mathbb{Z}_2$ je notranja operacija na vsaki podmnožici v $V^n$, ki vsebuje ničelno besedo. Zaradi tega je dvojiška koda $C$ linearna natanko takrat, ko je skupaj s kodnima besedama $u$ in $v$ tudi $u+v$ kodna beseda. Če namreč linearna koda $C$ vsebuje vsaj eno kodno besedo, imenujmo jo $u$, vsebuje C tudi ničelno besedo, saj je $u+u=0$. Ravno tako iz Lagrangevega izreka, ki pravi, da moč podgrupe deli moč grupe, sledi, da ima linearna koda $C$ $2^k$ elementov, kjer je $k \leq n$. Eksponentu $k$ pravimo *razsežnost* kode. Za linearne kode zapišemo Singletonovo mejo še preprosteje:

**Izrek 2.2.2.** [**Singletonova meja za linearne kode**] *Naj bo $V$ linearna koda dimenzije $k$. Tedaj velja $\delta \leq n-k+1$.* □

Definirajmo še pojem minimalne teže kode.

**Definicija 2.2.3. Minimalna teža kode** *je najmanjša teža, ki jo ima katera izmed kodnih besed.*

Minimalno težo linearne kode nam dá naslednja trditev:

**Trditev 2.2.4.** *Minimalna teža linearne kode je enaka njeni minimani razdalji.*

**Dokaz.** Iz definicij minimalne teže in minimalne razdalje vidimo, da minimalna razdalja ne more biti večja od minimalne teže.
Vzemimo sedaj kodni besedi $u$ in $v$, ki realizirata minimalno razdaljo: $d_H(u,v)=\delta$. Ta razdalja se ne spremeni, če obema besedama prištejemo isto besedo:

$$\delta = d_H(u,v) = d_H(u+u, v+u) = d_H(0, v+u).$$

Zaradi linearnosti kode je tudi $u+v$ kodna beseda. To pa pomeni, da znaša $d_H(0,v+u)$ vsaj toliko kot je minimalna teža kode. S tem je trditev dokazana. □

# Poglavje 3

# Kodiranje

Najbolj pomembna ter najpogosteje omenjana lastnost konvolucijskih kod je dejstvo, da kodne besede nimajo konstantne dolžine kot je to primer pri bločnih kodah. Druga pomembna lastnost konvolucijskih kod je ta, da $n$-terica, ki jo dobimo po kodiranju ni odvisna je od $k$-terice $m$, ki predstavlja sporočilo, temveč tudi od nekaterih $k$-teric, ki smo jih kodirali pred $m$. Zato rečemo, da ima kodirnik spomin.

Kot smo že uvodoma povedali, bomo opazovali le konvolucijske kode nad binarno abecedo $\mathbb{F}_2$. Zato bodo vse računske operacije mišljene kot operacije po modulu 2. Ko bomo rekli seštevanje, bomo mislili seštevanje po modulu 2. Enako velja za množenje.

## 3.1 Pomični registri

Vsakič, ko želimo narediti določene operacije nad nekim zaporedjem (vhodnih) podatkov, si postavimo nekaj vprašanj. Zanima nas predvsem, kako učinkovito, lahko predstavljivo in strojno uresničljivo znamo predstaviti orodje, ki bo izvajalo potrebne operacije. Pri različnih manipulacijah z zaporedji ponavadi uporabljamo *pomične registre*.

**Definicija 3.1.1. Pomični register z $m$ stanji** *je naprava, ki je sestavljena iz $m$ enot (ali flip-flopov) in ureja nadzor premikanja podatkov. Vsaka od enot ima en vhod in en izhod. V vsaki enoti časa se v registru opravijo naslednje operacije:*

*(i) Novi bit sporočila se pojavi na vhodu in se operacije po modulu 2 izvršijo na bitih znotraj registra tako, da dobimo nov bit na izhodu.*

*(ii) Vsebina vseh enot se premakne za eno mesto v desno (z izjemo skrajno desne enote).*

*(iii) Novi bit na vhodu postane vsebina prve enote.*

Pomične registre lahko na zelo eleganten način predstavimo s pomočjo polinomov nad $\mathbb{F}_2$. Prostor polinomov z neodvisno spremenljivko $x$ s koeficienti iz obsega $\mathbb{F}_2$ označimo z $\mathbb{F}_2[x]$.

**Definicija 3.1.2.** *Polinomu $g(x) = 1 + g_1x + g_2x^2 + \ldots + g_mx^m \in \mathbb{F}_2[x]$ rečemo, da je* **generator** *pomičnega registra z $m$ stanji, če je $g_i = 1$ v primeru, ko vsebina $i$-te enote nastopa v seštevanju, potrebnemu za izračun izhodnega bita.*

Generator pomičnega registra je torej ključen podatek, ki ga potrebujemo pri transformaciji zaporedja vhodnih podatkov v zaporedje izhodnih podatkov. Spodnji izrek nam pove, po kakšnem receptu izračunamo zaporedje izhodnih podatkov:

**Izrek 3.1.3.** *Naj bo dan pomični register z generatorjem $g(x)$ stopnje $m$. Če tok vhodnih podatkov $u_0, u_1, u_2, \ldots$ opišemo s formalno potenčno vrsto*

$$u(x) = u_0 + u_1x + u_2x^2 + \cdots$$

*nad $\mathbb{F}_2$ in tok izhodnih podatkov $c_0, c_1, c_2, \ldots$ s potenčno vrsto*

$$c(x) = c_0 + c_1x + c_2x^2 + \ldots,$$

*nad $\mathbb{F}_2$, potem velja:*

$$c(x) = u(x)g(x).$$

Izrek pove, da je kodno sporočilo enako konvoluciji vhodnega sporočila in generatorskega polinoma, ob predpostavki, da vse omenjena količine ustrezno zapišemo kot zaporedja. Od tod tudi ime *konvolucijske kode*. Dokažimo zgornji izrek in tako preverimo veljavnost poimenovanja.

**Dokaz.** Radi bi dokazali, da pomični register res pravilno izvrši množenje polinomov. Drugače povedano, radi bi pokazali, da je tok izhodnih podatkov konvolucija toka vhodnih podatkov in zaporedja, ki ga tvorijo koficienti generatorskega polinoma pomičnega registra.

Naj bo $u(x) = u_0 + u_1x + \ldots + u_{k-1}x^k$ in $g(x) = g_0 + g_1x + \ldots + g_{m-1}x^m$. Tedaj je koeficient $c_t$ pri $x^t$ v $c(x) = u(x)g(x)$ enak

$$c_t = g_0u_t + g_1u_{t-1} + \ldots + g_tu_0, \text{ če } t \leq m-1$$

in

$$c_t = g_0u_t + g_1u_{t-1} + \ldots + g_{m-1}u_{t-m+1}, \text{ če } t > m-1.$$

Poenostavimo dokaz tako, da privzamemo, da je $u_t = 0$, če $t > k-1 = \deg(u(x))$.

V pomičnem registru z generatorskim polinomom $g(x)$ označimo vsebino $i$-te enote registra z $X_i(t)$. Tedaj je izhodni bit v času $t$ linearna kombinacija vsebin registrskih enot $X_i(t)$

$$c_t = g_0 X_0(t) + \ldots g_{m-1} X_{m-1}(t).$$

V času 0 je $X_0(0) = u_0$ in $X_1(0) = \ldots = X_{m-1}(0) = 0$. Od tod je $c_0 = g_0 u_0$.
Splošneje, v času $t$, $t \leq m-1$ je $X_0(t) = u_t$, $X_1(t) = u_{t-1}, \ldots, X_t(t) = u_0$, v ostalih registrskih enotah pa so ničle. Zato je

$$c_t = g_0 u_t + g_1 u_{t-1} + \ldots + g_t u_0 \quad \text{za } t \leq m-1.$$

V času $t > m-1$ imamo

$$X_0(t) = u_t, X_1(t) = u_{t-1}, \ldots, X_{m-1}(t) = u_{t-m+1}$$

in zato tudi

$$c_t = g_0 u_t + g_1 u_{t-1} + \ldots + g_{m-1} u_{t-m+1}, \; t > m-1.$$

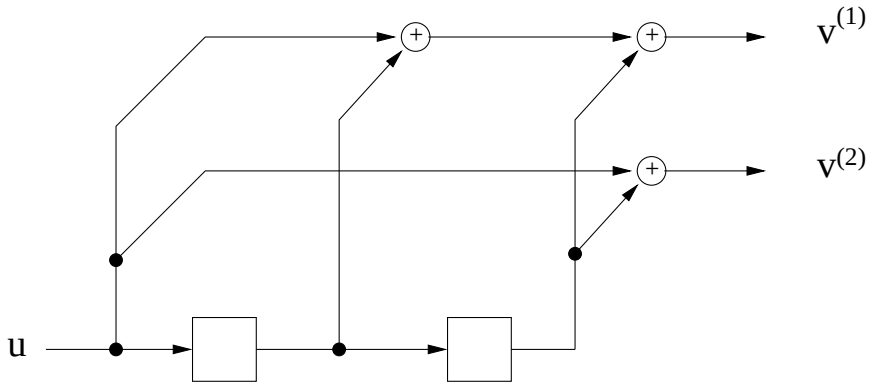
□

V grobem povedano, bomo konvolucijske kode dobili tako, da bo združili nekaj pomičnih registrov, njihove generatorske polinome pa zapisali v generatorsko matriko. Tudi brez stroge definicije pa iz pravkar dokazanega izreka lahko vidimo v uvodu omenjeno lastnost konvolucijskih kod: vhodni in izhodni podatki so lahko neskončni! Konvolucijske kode torej niso bločne kode. So jim pa podobne v tem, da imamo lahko konvolucijske kode za bločne nad določenimi neskončnimi obsegi.
Ilustrirajmo vsebino tega razdelka na primeru, ki ga bomo uporabljali tudi kasneje, ko bo to priročno.

**Primer:** Imejmo polinoma $g_1(D) = 1 + D + D^2$ in $g_2(D) = 1 + D^2$. Za vsak vhodni bit informacije bomo imeli dva izhodna bita kodnega sporočila. Združimo omenjena polinoma v generatorsko matriko

$$G_0(D) = \begin{pmatrix} 1 + D + D^2 & 1 + D^2 \end{pmatrix}$$

Premični register za dano matriko je predstavljen na sliki:




Slika 6 - Premični register za matriko $(1 + D + D^2\ 1 + D^2)$

Zakodirajmo zaporedje $\boldsymbol{u} = 110101$. Spodnja tabela kaže vsebino registra v različnih časovnih intervalih, časovne intervale ter izhodne bite.

| vsebina | čas $(t)$ | $v_t^{(1)}$ | $v_t^{(2)}$ |
|---|---|---|---|
| 100 | 0 | 1 | 1 |
| 110 | 1 | 0 | 1 |
| 011 | 2 | 0 | 1 |
| 101 | 3 | 0 | 0 |
| 010 | 4 | 1 | 0 |
| 101 | 5 | 0 | 0 |
| 010 | 6 | 1 | 0 |
| 001 | 7 | 1 | 1 |
| ?00 | 8 | | |

Na izhodu dobimo torej $\boldsymbol{v} = 1101010010001011$. Opazimo še, da smo zaporedje $\boldsymbol{u}$ terminirali z dvema dodatnima ničlama na vhodu, zato da bi pripeljali register v začetno ničelno stanje ali, bolje rečeno, da smo ga pripravili za kodiranje naslednjega sporočila. Zato znak ? v zadnji vrstici.

Poudarimo še, da smo tukaj uporabili spremenljivko $D$ namesto $x$. Razlogi za to spremembo bodo razvidni v četrtem poglavju.

$\Diamond$

## 3.2 Matematični model konvolucijskih kod

Za konvolucijski kodirnik s splošno informacijsko stopnjo $R = k/n, k \leq n$, vhodne podatke

$$\boldsymbol{u} = \boldsymbol{u}_0\boldsymbol{u}_1 \ldots = u_0^{(1)}u_0^{(2)} \ldots u_0^{(k)}u_1^{(1)}u_1^{(2)} \ldots u_1^{(k)} \ldots$$

zakodiramo v zaporedje

$$\boldsymbol{v} = \boldsymbol{v}_0\boldsymbol{v}_1 \ldots = v_0^{(1)}v_0^{(2)} \ldots v_0^{(n)}v_1^{(1)}v_1^{(2)} \ldots v_1^{(n)} \ldots,$$

kjer je

$$\boldsymbol{v}_t = f(\boldsymbol{u}_t, \boldsymbol{u}_{t-1}, \ldots \boldsymbol{u}_{t-m}).$$

Parametru $m$ pravimo *spomin kodirnika*. Za funkcijo $f$ zahtevamo, da je *linearna* funkcija iz $\mathbb{F}_2^{(m+1)k}$ v $\mathbb{F}_2^n$. Tako funkcijo lažje zapišemo v matrični obliki:

$$\boldsymbol{v}_t = \boldsymbol{u}_t G_0 + \boldsymbol{u}_{t-1} G_1 + \ldots \boldsymbol{u}_{t-m} G_m,$$

kjer je $G_i, 0 \leq i \leq m$ binarna $k \times n$ matrika. Iz zadnje enačbe sledi izraz

$$\boldsymbol{v}_0\boldsymbol{v}_1 \ldots = (\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots)G$$

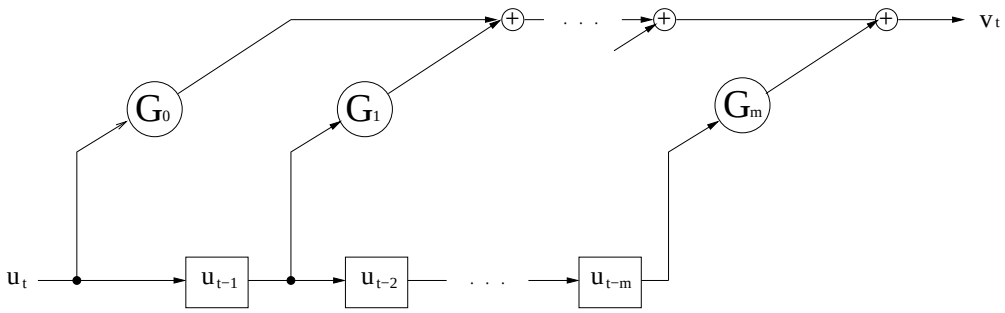
ali krajše

$$\boldsymbol{v} = \boldsymbol{u}G,$$

kjer je

$$G = \begin{pmatrix} G_0 & G_1 & \ldots & G_m & & \\ & G_0 & G_1 & \ldots & G_m & \\ & & \ddots & \ddots & & \ddots \end{pmatrix}$$

in kjer je v vseh manjkajočih delih matrike mišljena ničelna matrika ustrezne velikosti. Matriki $G$ pravimo *generatorska matrika*, podmatrikam $G_i, 0 \leq i \leq m$ pa *generatorske podmatrike*. Splošni konvolucijski pomični register lahko grafično predstavimo takole:




Slika 7 - Splošni konvolucijski pomični register

Za konec pa dodajmo še izrek, ki očitno sledi iz pravkar prikazanega matematičnega modela.

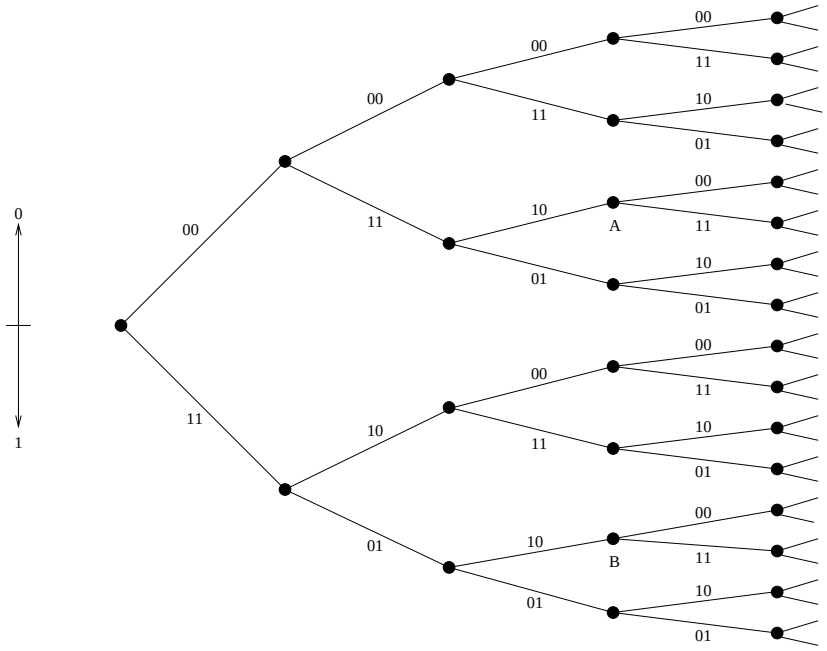
**Izrek 3.2.1.** *Konvolucijska koda je linearna.* □

## 3.3 Grafična predstavitev konvolucijskih kod

Zelo pogoste in zelo nazorne so različne grafične predstavitve konvolucijskih kod. Tukaj bomo predstavili tri različne grafične upodobitve konvolucijskih kod: drevesno predstavitev, predstavitev s pomočjo diagrama stanj in mrežno (brajdno) predstavitev.

### Drevesna predstavitev

*Drevesna predstavitev* je najosnovnejša grafična predstavitev konvolucijskih kod in nam ponuja temelj za izpeljavo nekaterih lastnosti konvolucijskih kod ter s tem tudi bolj kompaktne in bolj uporabne predstavitve. Ta predstavitev se, kot ena izmed možnih, ponuja iz dejstva, da je kodne besede izredno prikladno predstavljati v drevesni obliki. Najbolj levemu vozlu v taki predstavitvi pravimo *koren*. Kodirnik na vhodu dobi vedno en bit informacije, zatorej iz korena lahko gremo v dveh smereh, odvisno od vhodnega bita. V našem primeru bomo šli navzgor, ko je vhodni bit enak 0 ter navzdol, ko je vhodni bit enak 1. Na vsaki veji drevesa pa bomo imeli toliko binarnih cifer, kolikor jih dana koda proizvede za en vhodni bit informacije. Spodnja slika kaže eno od takšnih predstavitev.



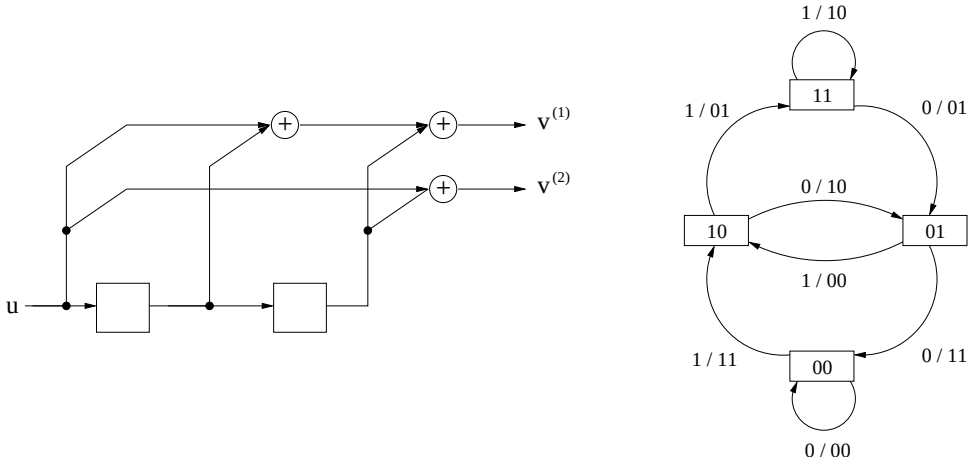

Slika 8 - Drevesna predstavitev konvolucijske kode

### Diagram stanj

*Stanje* kodirnika je opis prejetih bitov, torej nekakšen opis preteklosti za komunikacijski sistem. Tukaj je potrebno poudariti, da ob poznavanju stanja sistema ter prihajajočih bitov lahko brez težav določimo trenutne ter bodoče izhodne bite. Če ima naš kodirnik spomin velikosti $m$, v splošnem predstavimo stanje kodirnika v času $t$ z

$$\sigma_{\mathbf{t}} = u_{t-1}u_{t-2}\ldots u_{t-m}.$$

Iz vsega povedanega je tudi očtino, da ima kodirnik s spominom velikosti $m$ natanko $2^m$ med seboj različnih stanj. To pa tudi pomeni, da potrebujemo natanko $m$ vhodnih bitov, da pridemo v katerokoli možno stanje. Omenjeni definicija stanja sistema in dejstvo o prehodih nam omogočata uporabo t.i. *diagrama stanj*. Sledeča slika prikazuje konvolucijski kodirnik in ustrezni diagram stanj.

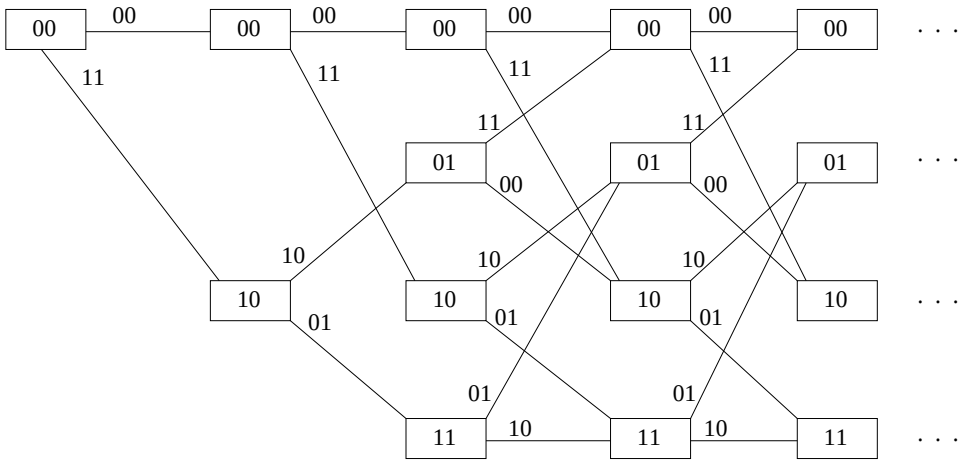



Slika 9 - Diagram stanj konvolucijske kode

### Mrežna predstavitev

Vrnimo se zopet k drevesni predstavitvi konvolucijske kode. Pozorno opazovanje drevesa nam razkrije, da primerno izbrani vhodni zaporedji pripeljeta kodirnik v isto stanje ter da sta ustrezni poddrevesi identični. Zato ni razloga, da ju opazujemo ločeno. Lahko ju kar zamenjamo z enim vozlom, ki ustreza danemu stanju v točno določenem času. Za vsako časovno enoto (ali *globino*) lahko ekvivalentne vozle zamenjamo z enim samim vozlom. Na ta način dobimo *mrežno predstavitev*, v kateri povezave navzgor in navzdol ustrezajo vhodnim informacijskim bitom 0 in 1 respektivno. Za nekoliko jasnejšo predstavitev poskrbi tudi spodnja slika, ki

prikazuje mrežno predstavitev za konvolucijski kodirnik, uporabljen pri ilustraciji diagrama stanj.




Slika 10 - Mrežna predstavitev konvolucijske kode

Iz same konstrukcije mrežne predstavitve je očitno, da vhodna zaporedja lahko preberemo iz dane strukture ravno tako kot jih na preprost način prebiramo iz kodnega drevesa. Mrežna predstavitev ima prednost predvsem zaradi dejstva, da ne zahteva veliko prostora za razliko od drevesne predstavitve, ki kmalu postane nepregledna. Mrežno predstavitev pa lahko brez težav izpeljemo tudi iz diagrama stanj.

Podajmo še primer.

**Primer:** Zakodirajmo sporočilo $\boldsymbol{u} = 1011\ldots$ z uporabo pravkar vpeljanih grafičnih predstavitev na zgornji sliki. V vseh primer dobimo kodno sporočilo $11100001\ldots$. Iz tega lahko sklepamo, da so vse predstavitve konsistentne. $\Diamond$

Omenimo še, da je mrežna predstavitev vsekakor najbolj uporabna, saj pride zelo prav pri odkodiranju kot bomo videli pri definiciji Viterbijevega algoritma.

## 3.4 Lastnosti razdalje

V tem razdelku si bomo najprej ogledali različne razdalje, ki jih lahko uporabljamo pri obravnavi konvolucijskih kod. Nato bomo s pridobljenim znanjem izpeljali nekatere pojme in orodja, ki nam pomagajo pri raziskovanju različnih lastnosti konvolucijskih kod ter izpeljavi učinkovitosti in ocen za verjetnosti bitnih napak pri praktični uporabi.

Imejmo binarno konvolucijsko kodo z informacijsko stopnjo $R = k/n$, racionalno generatorsko matriko $G(D)$ in spominom $m$. Informacijsko zaporedje

$$\boldsymbol{u}(D) = \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + \cdots$$

zakodiramo v kodno besedo

$$\boldsymbol{v}(D) = \boldsymbol{v}_0 + \boldsymbol{v}_1 D + \boldsymbol{v}_2 D^2 + \cdots ,$$

kjer je

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D).$$

Zaradi enostavnosti pa pišemo tudi $\boldsymbol{u} = \boldsymbol{u}_0\boldsymbol{u}_1 \ldots$ in $\boldsymbol{v} = \boldsymbol{v}_0\boldsymbol{v}_1 \ldots$ namesto $\boldsymbol{u}(D)$ in $\boldsymbol{v}(D)$. Zaradi lažjega razumevanja pojmov, ki jih bomo definirali spodaj, vpeljimo še oznako $\boldsymbol{x}_{[0,\,j]}$, ki predstavlja polinom $\boldsymbol{x}$, odrezan po $j$-tem členu oz. ustrezno zaporedje od člena $\boldsymbol{u}_0$ do členа $\boldsymbol{u}_j$. Zdaj lahko definiramo najosnovnejšo mero za razdaljo pri konvolucijskih kodah, ki ji pravimo *vrstična razdalja*.

**Definicija 3.4.1.** *Naj bo $C$ konvolucijska koda z generatorsko matriko $G(D)$.* **Vrstična razdalja $j$-tega reda** *$d_j^c$ generatorske matrike $G(D)$ je najmanjša Hammingova razdalja med dvema kodnima zaporedjema $\boldsymbol{v}_{[0,\,j]}$, ki izhajata iz informacijskih zaporedij $\boldsymbol{u}_{[0,\,j]}$ z različnima $\boldsymbol{u}_0$.*

Ker je konvolucijska koda linearna, vidimo, da je $d_j^c$ tudi najmanjša Hammingova teža poti $\boldsymbol{v}_{[0,\,j]}$, ki izhaja iz informacijskega zaporedja, za katerega je $\boldsymbol{u}_0 \neq 0$. Torej lahko pišemo

$$d_j^c = \min_{\boldsymbol{u}_0 \neq 0} \{w_H(\boldsymbol{v}_{[0,\,j]})\},$$

kjer $w_H()$ označuje Hammingovo težo zaporedja.
Naj bo zdaj

$$G(D) = G_0 + G_1 D + \ldots + G_m D^m$$

polinomska generatorska matrika s spominom m in naj bo ustrezna polneskončna matrika $\boldsymbol{G}$

$$\boldsymbol{G} = \begin{pmatrix} G_0 & G_1 & \ldots & G_m & & \\ & G_0 & G_1 & \ldots & G_m & \\ & & \ddots & \ddots & & \ddots \end{pmatrix}$$

kjer so $G_i$ binarne matrike velikosti $k \times n$ za $i \in \{0,\, \ldots,\, m\}$.
Označimo z $\boldsymbol{G}_j^c$ matriko, ki jo dobimo tako, da matriko $\boldsymbol{G}$ odrežemo po $(j+1)$-vem stolpcu, torej

$$\boldsymbol{G}_j^c = \begin{pmatrix} G_0 & G_1 & G_2 & \dots & G_j \\ & G_0 & G_1 & & G_{j-1} \\ & & G_0 & & G_{j-2} \\ & & & \ddots & \vdots \\ & & & & G_0 \end{pmatrix}$$

kjer privzamemo, da je $G_i = \mathbf{0}$, ko $i > m$.
S pravkar vpeljanimi oznakami lahko zapišemo

$$d_j^c = \min_{\boldsymbol{u} \neq 0} \{ w_H(\boldsymbol{u}_{[0,\,j]} \boldsymbol{G}_j^c) \}$$

Sklepamo lahko, da za izračun vrstične razdalje $j$-tega reda polinomske generatorske matrike zadošča že matrika, ki jo dobimo, če začetno matriko $\boldsymbol{G}$ odrežemo po $(j+1)$-vem stolpcu.
Potrebno je poudariti, da je vrstična razdalja lastnost kodirnika in ne kode same. Velja pa naslednji izrek.

**Izrek 3.4.2.** *Vrstična razdalja je invariantna v razredu ekvivalentnih kodirnih matrik.*

**Dokaz.** Naj bo $C_{cd}$ množica kodnih besed brez zamika:

$$C_{cd} = \{ \boldsymbol{v} \in C \mid \boldsymbol{v}_i = \mathbf{0}, i < 0 \; in \; \boldsymbol{v}_0 \neq \mathbf{0} \}$$

Množica $C_{cd}$ je podmnožica v $C$, ni pa podkoda, saj ni zaprta za seštevanje. Odvisna je samo od $C$ in neodvisna od izbrane generatorske matrike. Trditev izreka sledi iz dejstva, da je za kodirne matrike iskanje minimuma za $d_j^c = \min_{\boldsymbol{u}_0 \neq 0} \{ w_H(\boldsymbol{v}_{[0,\,j]}) \}$ enako kot iskanje minimuma za $\boldsymbol{v}_{[0,\,j]} \dim \boldsymbol{v} \in C_{cd}$. □

Iz pravkar dokazanega izreka očitno sledi

**Izrek 3.4.3.** *Naj bo $C$ konvolucijska koda. Vrstična razdalja $j$-tega reda kode $C$ je vrstična razdalja $j$-tega reda katerekoli kodirne matrike za $C$.*

Vrstično razdaljo $m$-tega reda $d_m^c$ racionalne generatorske matrike s spominom $m$ včasih imenujemo *minimalna razdalja* (generatorske matrike) in jo označimo z $d_{min}$. Minimalna razdalja določa sposobnost odkodirnika za popravljanje napak. Preden se lotimo te sposobnosti, pa si oglejmo še eno izmed lastnosti generatorskih matrik, ki jo dobimo s pomočjo vrstičnih razdalj.

**Definicija 3.4.4.** *Naj bo $G(D)$ racionalna generatorska matrika s spominom $m$. Vektorju, predstavljenem z $(m+1)$-terico*

$$\boldsymbol{d}^p = (d_0^c, d_1^c, \ldots, d_m^c),$$

*kjer je* $d_j^c, 0 \leq j \leq m$, *vrstična razdalja* $j$*-tega reda, imenujemo* **razdaljni profil generatorske matrike** $G(D)$.

Tako definiran razdaljni profil je lastnost generatorske matrike. Vendar pa, zaradi dejstva, da je vrstična razdalja $j$-tega reda ista za vse ekvivalentne kodirne matrike ter (kot bomo videli kasneje) da je spomin enako velik za ekvivalentne minimalno-bazične (kanonične) kodirne matrike, lahko vpeljemo tudi pojem razdaljnega profila kode

**Definicija 3.4.5.** *Naj bo* $C$ *konvolucijska koda z minimalno-bazično kodirno matriko* $G_{mb}(D)$ *s spominom* $m$. $(m+1)$*-erica*

$$\boldsymbol{d}^p = (d_0^c, d_1^c, \ldots, d_m^c),$$

*kjer je* $d_j^c, 0 \leq j \leq m$, *vrstična razdalja* $j$*-tega reda matrike* $G_{mb}$, *se imenuje* **razdaljni profil kode** $C$.

Razdaljne profile kod lahko primerjamo med seboj. Zato pravimo, da je pri generatorski matriki s spominom $m$ razdaljni profil $\boldsymbol{d}^p$ *boljši* od razdaljnega profila $\boldsymbol{d}^{p'}$ neke druge generatorske matrike z enako informacijsko stopnjo $R$ in spominom enake velikosti $m$, kadar obstaja tak $\ell$, da je

$$d_j^c \begin{cases} = d_j^{c'}, & j = 0, 1, \ldots, \ell - 1 \\ > d_j^{c'}, & j = \ell \end{cases}$$

Skladno s tako definicijo primerave med različnimi razdaljnimi profili pravimo, da ima konvolucijska koda $C$ *optimalni razdaljni profil*, kadar ne obstaja generatorska matrika z isto informacijsko stopnjo in spominom, ki bi imela boljši razdaljni profil. Generatorska matrika z optimalnim razdaljnim profilom je kodirna matrika.

Ob koncu razdelka bomo potrebovali še en izrek, ki nam pove nekaj o lastnostih vrstičnih razdalj. Izreka ne bomo dokazovali, saj je precej očiten.

**Izrek 3.4.6.** *Vrstične razdalje generatorske matrike zadoščajo naslednjim pogojem:*

*(i)* $d_j^c \leq d_{j+1}^c, j = 0, 1, 2, \ldots$

*(ii) Zaporedje* $d_0^c, d_1^c, d_2^c, \ldots$ *je omejeno navzgor.*

*(iii) Zaporedje* $d_j^c$ *postane stacionarno, ko* $j$ *narašča.* □

Iz izreka vidimo, da tvorijo vrstične razdalje $d_j^c$ nenaraščajočo funkcijo spremenljivke $j$. Pravimo ji *funkcija vrstičnih razdalj*. Vemo tudi, da za to funkcijo obstaja limita

$$d_\infty^c = \lim_{j\to\infty} d_j^c$$

in velja

$$d_0^c \leq d_1^c \leq \ldots \leq d_\infty^c.$$

Ponovimo še enkrat definicijo proste razdalje, ki smo jo podali na samem začetku, le da uporabimo oznake tega razdelka.

**Definicija 3.4.7.** *Naj bo $C$ konvolucijska koda. Najmanjši Hammingovi razdalji med dvema različnima kodnima besedama*

$$d_{free} = \min_{\boldsymbol{v}\neq\boldsymbol{v}'}\{d_H(\boldsymbol{v},\boldsymbol{v}')\}$$

*pravimo* **prosta razdalja** *kode $C$.*

Iz linearnosti konvolucijskih kod sklepamo, da je $d_{free}$ tudi najmanjša Hammingova teža neničelnih kodnih besed. Poudarimo, da je prosta razdalja lastnost kode.

Prosta razdalja je glavna mera za sposobnost popravljanja napak, kadar komunikacija poteka preko kanala z majhno verjetnostjo napak in uporabljamo odkodiranje po principu najverjetnejšega kanditata.

Označimo z $E_t$ množico vseh vzorcev, ki vsebujejo kvečjemu $t$ napak. Iz izreka o Singletonovi meji sledi naslednji izrek.

**Izrek 3.4.8.** *Konvolucijska koda $C$ lahko popravi vse vzorce iz $E_t$ natanko takrat, ko je $d_{free} > 2t$.* □

Seveda obstaja povezava med vrstičnimi razdaljami in prosto razdaljo. Formuliramo jo z sledečim izrekom.

**Izrek 3.4.9.** *Za vsako konvolucijsko kodo $C$ velja:*

$$d_{free} = d_\infty^c.$$

**Dokaz.** Iz izreka 2.6.6 vemo, da obstaja tako naravno število $k$, da je

$$d_k^c = d_{k+1}^c = \ldots = d_\infty^c.$$

Obstaja tudi kodna beseda teže $d_k^c$. Naj bo $G(D)$ polinomska kodirna matrika kode $C$. Po definiciji $d_k^c$ izhaja kodno zaporedje $\boldsymbol{v}_{[0,\,k]}$ iz informacijskega zaporedja $\boldsymbol{u}_{[0,\,k]}$

z $\boldsymbol{u}_0 \neq \mathbf{0}$, tako da velja $w_H(\boldsymbol{u}_{[0,\,k]}) = d_k^c$. Ker ima $G_0 = G(0)$ poln rang, lahko za vsak $i > k$ izberemo takšen $\boldsymbol{u}_i$, da je

$$\boldsymbol{u}_i G_0 + \boldsymbol{u}_{i-1} G_1 + \ldots + \boldsymbol{u}_{i-m} G_m = \mathbf{0},$$

kjer je $\boldsymbol{u}_n = \mathbf{0}$ za $n < 0$. Potem je

$$(\boldsymbol{u}_{[0,\,k]} \boldsymbol{u}_{k+1} \boldsymbol{u}_{k+2} \ldots)\boldsymbol{G} = (\boldsymbol{v}_{[0,\,k]} \mathbf{00} \ldots) \in C$$

Od tod sklepamo, da je

$$w_H(\boldsymbol{v}_{[0,\,k]} \mathbf{00} \ldots) = w_H(\boldsymbol{v}_{[0,\,k]}) = d_k^c$$

Zato velja tudi

$$d_{free} \leq d_k^c = d_\infty^c.$$

Lahko predpostavimo, da v $C$ obstaja kodna beseda $\boldsymbol{v} = \boldsymbol{v}_0 \boldsymbol{v}_1 \ldots$ s težo $d_{free}$ in $\boldsymbol{v}_0 \neq \mathbf{0}$. Za vse $j$ imamo

$$d_j^c \leq w_H(\boldsymbol{v}_0 \boldsymbol{v}_1 \ldots \boldsymbol{v}_j) \leq d_{free}$$

Če pa postavimo $j \geq k$, smo izrek dokazali. □

## 3.5 Rodovna funkcija konvolucijske kode

Pogosto potrebujemo nekoliko globlje poznavanje strukture razdalj neke konvolucijske kode. V ta namen vpeljemo pojme *rodovne funkcije* in *števcev poti*.

Označimo z $n_{d_{free}+i}$ število poti s težo $d_{free} + i$, ki na samem začetku zapustijo ničelno pot in se ne vrnejo nanjo do svojega konca. Številu $n_{d_{free}+i}$ pravimo $(i+1)$-va *spektralna komponenta*. Zaporedju

$$n_{d_{free}+i}, i = 0, 1, 2, \ldots$$

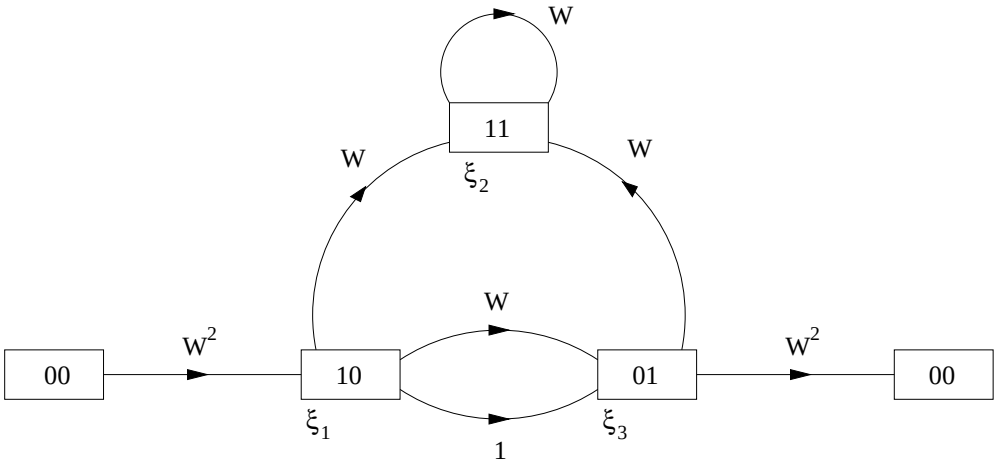
pravimo *težni spekter* kode. Rodovni funkciji težnega spektra,

$$T(W) = \sum_{i=0}^{\infty} n_{d_{free}+i} W^{d_{free}+i}$$

previmo *števec tež poti*. Tej funkciji in nekaterim njenim izpeljankam bomo posvetili večino tega razdelka.

Težni spekter in njemu pridružen razdaljni spekter lahko dobimo na preprost način iz diagrama stanj dane konvolucijske kode. Če iz ničelnega stanja umaknemo zanko in namesto nje narišemo diagram stanj, ki ima ločeni vhodno in izhodno ničelno stanje, lahko preštevanje poti precej poenostavimo in matematično bolj nazorno predstavimo. To metodo preštevanja poti je iznašel Viterbi (glej v [15]) in jo bomo zaradi lažjega razumevanja predstavili kar na primeru.

Za primer izberimo konvolucijski kodirnik in diagram stanj, podana v razdelku 2.5.2. Kot že prej opisano, v diagramu stanj izbrišemo zanko v ničelnem stanju in ničelno stanje razcepimo na vhodno in izhodno ničelno stanje. Povezave označimo z $W^0 = 1, W$ in $W^2$, kjer eksponent ustreza teži posamezne povezave. Kot rezultat dobimo t.i. *sliko poteka signala*. V našem primeru dobimo




Slika 11 - Potek signala

Naj bo zdaj na vhodu (vhodnem ničelnem stanju) vhodni bit enak 1. S $T(W)$ označimo rodovno funkcijo za težo poti $W$. Torej je to naš števec tež poti, kot je bil definiran zgoraj. Naj označujejo $\xi_1, \xi_2$ in $\xi_3$ spremenljivke, ki predstavljajo teže vseh poti od vhodnega ničelnega stanja do vmesnih stanj na diagramu stanj. Iz prejšnje slike preberemo naslednji sistem linearnih enačb

$$\xi_1 = \xi_3 + W^2,$$

$$\xi_2 = W\xi_1 + W\xi_2,$$

$$\xi_3 = W\xi_1 + W\xi_2,$$

in

$$T(W) = W^2\xi_3.$$

Zgornje enačbe lahko prepišemo v matrično obliko

$$\begin{pmatrix} 1 & 0 & -1 \\ -W & 1-W & 0 \\ -W & -W & 1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} W^2 \\ 0 \\ 0 \end{pmatrix}$$

Z uporabo Cramerjevega pravila dobimo

$$\xi_3 = W^3/(1-2W)$$

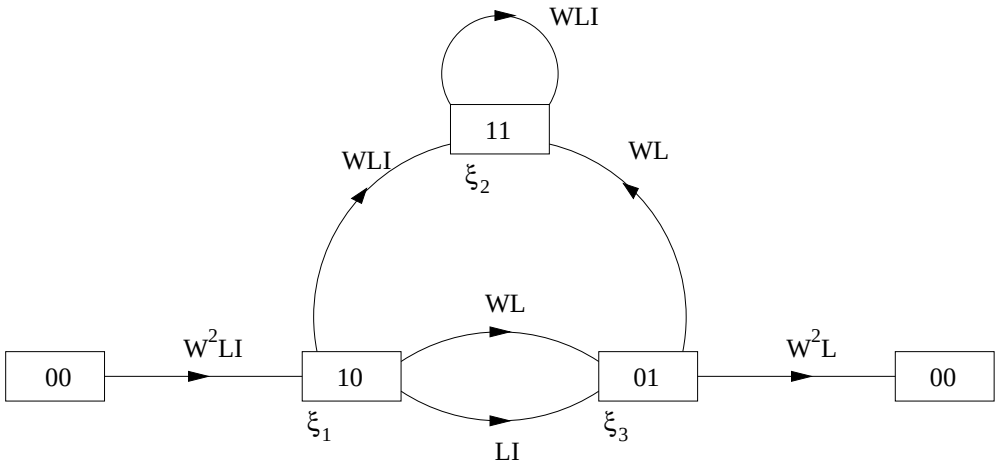
Zato je

$$T(W) = W^5/(1-2W) = W^5 + 2W^6 + 4W^7 + \ldots + 2^k W^{k+5} + \ldots$$

Od tod pa preberemo, da je prosta razdalja $d_{free} = 5$ in da so spektralne komponente $n_{d_{free}+i}, i = 0, 1, 2, \ldots$ enake 1,2,4, ....

Viterbi je sliko poteka signala uporabil tudi pri izpeljavi *razširjenega števca poti*, ki poleg štetja poti, glede na njihovo težo, šteje tudi poti glede na njihovo dolžino in število enic v informacijskem zaporedju. Dolžino poti bomo označili z $L$ in število enic z $I$.

Vrnimo se k začetni sliki in označimo povezave ne samo z $W^w$, kjer je $w$ teža povezave, temveč jih označimo tudi z $LI^i$, kjer je $i$ število enic v informacijskem zaporedju, ki ustreza posamezni veji. Tako dobimo *razširjeno sliko poteka signala*




Slika 12 - Razširjeni potek signala

Iz njega lahko preberemo sistem linearnih enačb

$$\begin{pmatrix} 1 & 0 & -LI \\ -WLI & 1-WLI & 0 \\ -WL & -WL & 1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \begin{pmatrix} W^2LI \\ 0 \\ 0 \end{pmatrix}$$

in

$$T(W, L, I) = W^2 L \xi_3.$$

Če rešimo zgornji sistem in njegovo rešitev vstavimo v rodovno funkcijo, dobimo

$$T(W, L, I) = \frac{W^5 L^3 I}{1 - WL(1 + L)I} = W^5 L^3 I + W^6 L^4 (1 + L) I^2 + W^7 L^5 (1 + L)^2 I^3 + \ldots$$

(Seštevamo člene oblike $W^{5+k} L^{3+k} (1 + L)^k I^{1+k}$.)

Števec tež poti je lastnost kode, medtem ko je razširjeni števec poleg kode odvisen tudi od preslikave med informacijskimi in kodnimi simboli, torej odvisen je kodirne matrike.

Števci poti in tež poti so zelo koristen pripomoček za ocenjevanje in izpeljevanje različnih mej za napake pri odkodiranju z uporabo metode najverjetnejšega kandidata. Tudi mi jih bomo uporabili v petem poglavju, kjer se bomo ukvarjali z odkodiranjem konvolucijskih kod.

S tem je poglavje, ki opisuje konvolucijske kode in nekatere njihove lastnosti z matematičnega stališča končano. V naslednjem poglavju bo naša pozornost usmerjena v raziskovanje generatorskih matrik in ustreznih kodirnikov. Poskusili bomo izpeljati kriterije, ki nam bodo iz različnih razredov matrik pomagali poiskati takšne matrike, za katere bomo vedeli, da imajo nekatere prednosti pri praktični izvedbi konvolucijskih kod.

# Poglavje 4

# Algebrska struktura generatorskih matrik

## 4.1 Splošni pojmi

V poglavju *Matematični model konvolucijskih kod* smo prišli do matematičnega opisa konvolucijskih kod, kjer so glavne vloge pripadle generatorski matriki in konvoluciji zaporedij. Tukaj se bomo osredotočili na konvolucijsko matriko ter iz nekaterih njenih algebraičnih lastnosti izpeljali nekaj karakteristik, ki nam bodo pomagale pri izbiri optimalnih generatorjev in tako predstavili tudi nekaj koristnih napotkov za konstrukcije optimalnih konvolucijskih kod.

Za raziskovanje algebrske strukture konvolucijskih kod bomo uporabljali pojem *modula*. Modul je struktura podobna vektorskemu prostoru, le da je definiran nad kolobarjem (vektorski prostor pa je definiran nad poljem).

**Definicija 4.1.1.** *Naj bo $R$ kolobar z enoto $1$. Njegove elemente imenujmo* **skalarje**. *Imejmo še množico $M$ z operacijama seštevanja (označimo jo s $+$) in množenja s skalarjem (označimo jo z $\cdot$). Seštevanje je notranja operacija (dvema elementoma množice $M$ priredi element množice $M$), množenje pa zunanja (elementu množice $M$ in elementu kolobarja $R$ priredi element množice $M$). Množica $M$ je* **modul nad kolobarjem** *$R$, če $M$ skupaj z obema operacijama zadošča naslednjim pogojem:*

*1. $(M, +)$ je Abelova grupa.*

*2. Distributivnost: Za poljubna elementa $m_1, m_2 \in M$ in poljuben skalar $r \in R$ velja*

$$r \cdot (m_1 + m_2) = r \cdot m_1 + r \cdot m_2.$$

*3. Distributivnost: Za poljuben element $m \in M$ velja $1 \cdot m = m$ in*

$$(r_1 + r_2) \cdot m = r_1 \cdot m + r_2 \cdot m,$$

*za poljubna skalarja $r_1, r_2 \in R$.*

*4. Asociativnost: Za poljubne $m \in M$ in $r_1, r_2 \in R$ velja*

$$(r_1 \cdot r_2) \cdot m = r_1 \cdot (r_2 \cdot m).$$

**Opomba:** V prihodnje bomo znak $\cdot$ izpustili in namesto $r \cdot m$ pisali kar $rm$.

Iz definicije modula sledi, da je vsak modul, definiran nad poljem (namesto nad kolobarjem), dejansko vektorski prostor. Modul je torej bolj splošna algebrska struktura z nekoliko šibkejšimi lastnostmi.

Najprej se spomnimo Laplaceove formule za razvoj determinante matrike, ki pravi, da je za poljubno matriko $T$ njena determinanta enaka

$$\det T = \sum_{\ell=1}^{n} t_{i\ell} C_{i\ell},$$

kjer je $t_{i\ell}$ $(i,\ \ell)$-ti element matrike $T$ in $C_{i\ell}$ kofaktor za $t_{i\ell}$ (se pravi, če ne upoštevamo predznaka $C_{i\ell}$, lahko rečemo, da je $C_{i\ell}$ minor $T$). Mi bomo obravnavali matrike, katerih elementi bodo iz kolobarja polinomov. Kot spremenljivko bomo v naslednjem razdelku vpeljali *operator zamika* $D$.

Iz Laplaceovega razvoja sledi, da je vsak skupni polinomski faktor vseh minorjev matrike $T(D)$ tudi polinomski faktor determinante $\det T(D)$. Prav tako je vsak minor velikosti $i \times i$ determinanta in ga lahko izrazimo s pomočjo njegovih $(i-1) \times (i-1)$ minorjev. Torej je tudi največji skupni delitelj vseh determinant $i \times i$ podmatrik polinomski večkratnik največjega skupnega delitelja vseh determinant podmatrik velikosti $(i-1) \times (i-1)$. Omenjeni največji skupni delitelj determinant $i \times i$ podmatrik bomo odslej označevali z $\Delta_i(D)$. To velja tudi v primeru, ko originalna matrika ni kvadratna.

## 4.2 Konvolucijske kode in njihovi kodirniki

Pri neki informacijski stopnji $R = k/n, k \leq n$ se morata vhodno zaporedje

$$\boldsymbol{u} = \ldots u_{-1} u_0 u_1 u_2 \ldots,$$

kjer je $\boldsymbol{u}_i = u_i^{(1)} u_i^{(2)} \ldots u_i^{(k)},\ u_i^{(j)} \in \mathbb{F}_2$ za $1 \leq j \leq k,\ i \in \mathbb{Z}$, in izhodno zaporedje

$$\boldsymbol{v} = \ldots v_{-1} v_0 v_1 v_2 \ldots,$$

kjer je $\boldsymbol{v}_i = v_i^{(1)} v_i^{(2)} \ldots v_i^{(n)},\ v_i^{(j)} \in \mathbb{F}_2$ za $1 \leq j \leq k,\ i \in \mathbb{Z}$, začeti v nekem končnem (pozitivnem ali negativnem) času in se lahko končata ali ne. Najpogosteje to izrazimo s pomočjo operatorja zamika $D$ in pišemo

$$u^{(\ell)}(D) = \ldots + u_{-1}^{(\ell)} D^{-1} + u_0^{(\ell)} + u_1^{(\ell)} D^1 + u_2^{(\ell)} D^2 + \ldots$$

$$v^{(\ell)}(D) = \ldots + v_{-1}^{(\ell)} D^{-1} + v_0^{(\ell)} + v_1^{(\ell)} D^1 + v_2^{(\ell)} D^2 + \ldots$$

Oglejmo si zdaj matematično okolje, v katerem bomo delali.

Z $\mathbb{F}_2((D))$ označimo polje binarnih Laurentovih vrst. Vsak element

$$x(D) = \sum_{i=r}^{\infty} x_i D^i \in \mathbb{F}_2((D)),\ r \in \mathbb{Z},$$

vsebuje kvečjemu končno mnogo členov, v katerih $D$ nastopa v negativni potenci. "Časovnemu indeksu" $r$, v katerem se vrsta začne, pravimo *zamik Laurentove vrste* in ga označimo z $zam(x(D))$. Tako ima, na primer, vrsta

$$x(D) = D^{-2} + D + D^4 + D^{11} + \ldots$$

zamik enak $zam(x(D)) = -2$.
Označimo zdaj z $\mathbb{F}_2[[D]]$ kolobar formalnih potenčnih vrst. Element

$$f(D) = \sum_{i=0}^{\infty} f_i D^i \in \mathbb{F}_2[[D]]$$

je Laurentova vrsta brez členov z $D$-jem v negativni potenci. Torej je $\mathbb{F}_2[[D]]$ podmnožica v $\mathbb{F}_2((D))$. Elementu

$$f(D) = \sum_{i=0}^{\infty} f_i D^i$$

z $f_0 = 1$ rečemo, da je *brez zamika*. Polinomi oblike $p(D) = \sum_{i=0}^{\infty} p_i D^i$ imajo le končno mnogo pozitivnih potenc in so brez negativnih potenc. Torej je vsak polinom z $p_0 = 1$ brez zamika.
Množica binarnih polinomov $\mathbb{F}_2[D]$ je podmnožica v $\mathbb{F}_2[[D]]$ in zato tudi podmožica v $\mathbb{F}_2((D))$. Množenje polinomov v $\mathbb{F}_2[D]$ je navadno množenje polinomov z operacijami po modulu 2. $\mathbb{F}_2[D]$ ne more biti polje, ker ima samo polinom 1 multiplikativni inverz. Za poljuben par polinomov $x(D), y(D) \in \mathbb{F}_2[D]$, pri čemer je $y(D) \neq 0$, z deljenjem sestavimo element $x(D)/y(D)$. Vsi neničelni elementi oblike $x(D)/y(D)$ tvorijo množico $\mathbb{F}_2(D)$, ki je polje. Imenujemo ga *polje binarnih racionalnih funkcij* in je podpolje v $\mathbb{F}_2((D))$.
Oglejmo si še $s$-terice elementov iz $\mathbb{F}_2[D], \mathbb{F}_2[[D]], \mathbb{F}_2(D)$ oziroma $\mathbb{F}_2((D))$. Tako $s$-terico

$$\boldsymbol{x}(D) = (x^{(1)}(D), x^{(2)}(D), \ldots, x^{(s)}),$$

kjer so $x^{(1)}(D), x^{(2)}(D) \ldots x^{(s)}(D) \in \mathbb{F}_2((D))$, lahko zapišemo kot vektor

$$\boldsymbol{x}(D) = \sum_{i=r}^{\infty} (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(s)}) D^i, \ r \in \mathbb{Z},$$

kjer so komponente enake

$$x^{(j)}(D) = \sum_{i=r}^{\infty} x_i^{(j)} D^i, 1 \leq j \leq s.$$

Torej lahko $s$-terice elementov iz $\mathbb{F}_2((D))$ označimo z $\boldsymbol{\mathbb{F}_2^s((D))}$ in dobimo $s$-dimenzionalen vektorski prostor nad poljem binarnih Laurentovih vrst $\mathbb{F}_2((D))$. Na podoben način dobimo tudi $\mathbb{F}_2^s[D], \mathbb{F}_2^s[[D]], \mathbb{F}_2^s(D)$.
Elementu $\boldsymbol{x}(D) \in \mathbb{F}_2^n[D]$ rečemo polinom. **Stopnja elementa**

$$\boldsymbol{x}(D) = \sum_{i=0}^{m} (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}) D^i$$

je $m$, pri predpostavki, da $(x_m^{(1)}, x_m^{(2)}, \ldots, x_m^{(n)}) \neq (0, 0, \ldots, 0)$.

Vrnimo se zdaj h konvolucijskim kodam in uporabimo vpeljane oznake, kjer bo to primerno ter si oglejmo, kaj lahko izvemo z uporabo le-teh in nekaterih znanih dejstev o definiranih objektih.
Označimo z $\boldsymbol{u}(D) \in \mathbb{F}_2^k((D))$ vhodno zaporedje ter z $\boldsymbol{v}(D) \in \mathbb{F}_2^n((D))$ izhodno zaporedje za konvolucijski kodirnik z informacijsko stopnjo $R = k/n$. Vpeljimo še pojem *izvedljivosti*.

**Definicija 4.2.1.** *Racionalna funkcija $g(D) = f(D)/q(D)$ je* **izvedljiva**, *če je $q(D)$ brez zamika. Matriko $G(D)$, katere elementi so racionalne funkcije, imenujemo* **racionalna matrična funkcija prenosa**. *Racionalna matrična funkcija prenosa $G(D)$ za linearen sistem z več vhodi in/ali izhodi je* **izvedljiva**, *če so njeni elementi so izvedljive funkcije.*

Zdaj imamo vse potrebno za formalno definicijo *konvolucijske preslikave*, ki bo osnova za izpeljave različnih lastnosti konvolucijskih kod.

**Definicija 4.2.2. Konvolucijska preslikava** *z informacijsko stopnjo $R = k/n$ nad poljem racionalnih funkcij $\mathbb{F}_2(D)$ je linearna preslikava*

$$\begin{aligned} \tau : \mathbb{F}_2^k((D)) &\longrightarrow \mathbb{F}_2^n((D)) \\ \boldsymbol{u}(D) &\longmapsto \boldsymbol{v}(D), \end{aligned}$$

*ki jo lahko zapišemo kot*

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D),$$

*kjer je $G(D)$ $k \times n$ matrična funkcija prenosa z elementi iz $\mathbb{F}_2(D)$. Zaporedju $\boldsymbol{v}(D)$ pravimo* **kodno zaporedje**, *ki izhaja iz* **informacijskega zaporedja** $\boldsymbol{u}(D)$

Naravna zahteva, ki se ob tej definiciji ponuja sama od sebe, je ta, da moramo biti zmožni rekonstruirati informacijsko zaporedje $\boldsymbol{u}(D)$ iz kodnega zaporedja v primeru, ko v kanalu ni šuma. Sicer bi bil transducer neuporaben. Torej mora biti transducer injektiven, kar pa pomeni, da mora imeti matrika $G(D)$ rang $k$ nad poljem $\mathbb{F}_2(D)$.
Iz vsega povedanega je smiselna naslednja definicija.

**Definicija 4.2.3. Konvolucijska koda $\boldsymbol{C}$** *z informacijsko stopnjo $R = k/n$ nad $\mathbb{F}_2$ je slika konvolucijskega transducerja informacijske stopnje $R = k/n$, katerega matrična funkcija prenosa $G(D)$ je ranga $k$ na $\mathbb{F}_2(D)$.*

**Opomba.** Iz zgornje definicije sledi, da lahko na konvolucijsko kodo $C$ z informacijsko stopnjo $R = k/n$ nad $\mathbb{F}_2$ s $k \times n$ matriko prenosa ranga $k$ nad $\mathbb{F}_2(D)$ gledamo tudi kot na linearno bločno z informacijsko stopnjo $R = k/n$ nad (neskončnim) poljem Laurentovih vrst, zakodiranih z matriko $G(D)$.
Od tukaj naprej bomo opazovali le izvedljive matrične funkcije prenosa.

**Definicija 4.2.4.** *Matrično funkcijo prenosa (konvolucijske kode) imenujemo* **generatorska matrika**, *če je izvedljiva (in je polnega ranga).*

Skladno s to definicijo rečemo, da je izvedljiva matrična funkcija prenosa $G(D)$ *brez zamika*, če za vsaj enega od njenih elementov $f(D)/q(D)$ velja $f(0) \neq 0$. Če ima $G(D)$ zamik, potem jo lahko zapišemo kot

$$G(D) = D^i G_d(D),$$

kjer je $i \geq 1$ in $G_d(D)$ brez zamika.

**Izrek 4.2.5.** *Vsaka konvolucijska koda $C$ ima generatorsko matriko, ki je brez zamika.*

**Dokaz.** Naj bo $G(D)$ generatorska matrika za $C$. Njene neničelne elemente lahko zapišemo v obliki

$$g_{ij}(D) = D^{s_{ij}} f_{ij}(D)/q_{ij}(D),$$

kjer je $s_{ij}$ tako celo število, da velja $f_{ij}(0) = q_{ij}(0) = 1, 1 \leq i \leq k, 1 \leq j \leq n$. Torej je $s_{ij}$ zamik zaporedja

$$g_{ij}(D) = D^{s_{ij}} f_{ij}(D)/q_{ij}(D) = D^{s_{ij}} + g_{s_{ij}+1} D^{s_{ij}+1} + \ldots$$

Postavimo $s = \min\{s_{ij}\}$. Potem je

$$G'(D) = D^{-s}G(D)$$

izvedljiva in brez zamika. Ker pa je $D^{-s}$ skalar v $\mathbb{F}_2((D))$, $G(D)$ in $G'(D)$ generirata enako konvolucijsko kodo. Zatorej je $G'(D)$ generatorska matrika kode $C$ in je brez zamika. □

Pravkar dokazani izrek nam pove, da vsako konvolucijsko kodo lahko zakodiramo z več bistveno različnimi kodirniki. Doslej smo v glavnem operirali z racionalnimi funkcijami. Seveda se postavlja vprašanje, če je možno brez kakršnihkoli izgub prevesti dosedanja opažanja na polinome, ki nam, če ne drugega, olajšajo marsikateri izračun. Naslednji izrek nam bo pri tej prevedbi v veliko pomoč.

**Izrek 4.2.6.** *Vsaka konvolucijska koda $C$ ima polinomsko generatorsko matriko, ki je brez zamika.*

**Dokaz.** Naj bo $G(D)$ neka izvedljiva generatorska matrika konvolucijske kode $C$, ki je brez zamika. S $q(D)$ označimo najmanjši skupni večkratnik vseh imenovalcev v izrazu

$$g_{ij}(D) = D^{s_{ij}} f_{ij}(D)/q_{ij}(D).$$

Ker je $q(D)$ polinom brez zamika, je

$$G'(D) = q(d)G(D)$$

polinomska generatorska matrika kode $C$, ki je brez zamika. □

Kodirniku s polinomsko generatorsko matriko brez zamika pravimo *polinomski kodirnik*.

Seveda moramo biti pri izbiri generatorskih matrik izredno pozorni. Ne sme se pripetiti, da zaradi končnega števila napak pri prenosu sporočila prek kanala pri odkodiranju ustvarimo neskončno mnogo napak. V takem primeru bi nevede popolnoma narobe sklepali o sporočilu, ki nam je bilo poslano. Kaj lahko bi se zgodilo, da popolnoma razumljivo vhodno sporočilo odkodiramo v nesmiselno sporočilo. Da bi se takim situacijam izognili, najprej definirajmo "nezaželene" generatorske matrike.

**Definicija 4.2.7.** *Generatorska matrika konvolucijske kode je* **katastrofična**, *če obstaja tako vhodno informacijsko zaporedje $\boldsymbol{u}(D)$ z neskončno mnogo neničelnimi biti, $w_H(\boldsymbol{u}(D)) = \infty$, ki ga zakodiramo v kodne besede $\boldsymbol{v}(D)$ s končno mnogo neničelnimi biti, $w_H(\boldsymbol{v}(D)) < \infty$.*

Katastrofičnost je lastnost generatorske matrike. Vsaka konvolucijska koda ima tako katastrofične kot nekatastrofične generatorske matrike. Iz vsega povedanega sledi,

da je izbira generatorske matrike izrednega pomena pri sestavi kodirnika. Vpeljimo definicijo *kodirne matrike*.

**Definicija 4.2.8.** *Generatorski matriki $G(D)$, za katero je matrika $G(0)$ polnega ranga, pravimo* **kodirna matrika**.

Iz te definicije ter iz lastnosti, ki smo jih že izpeljali, sledi naslednji izrek.

**Izrek 4.2.9.** *Kodirna matrika je izvedljiva in brez zamika.* □

## 4.3 Smithova oblika polinomske konvolucijske generatorske matrike

Preden nadaljujemo, podajmo še eno definicijo, ki jo bomo potrebovali v nadaljevanju.

**Definicija 4.3.1.** *Kvadratno polinomsko matriko $T(x)$, katere determinanta je enaka nekemu neničelnemu elementu polja, imenujemo* **unimodularna matrika**.

Torej so elementi inverza poljubne unimodularne matrike polinomi. Poleg tega pa vsako polinomsko matriko $S(x)$, za katero poznamo produkt $S(x)T(x)$ z neko dano unimodularno, lahko pridobimo nazaj:

$$S(x) = [S(x)T(x)]T^{-1}(x).$$

Unimodularne $n \times n$ matrike so enote kolobarja matrik velikosti $n \times n$ nad $\mathbb{F}[x]$.

**Definicija 4.3.2.** *Za matriko $P(x)$ velikosti $k \times n$ in ranga $r$ bomo rekli, da je v* **Smithovi obliki**, *če velja*

$$P(x) = A(x)\Pi(x)B(x),$$

*kjer sta $A(x)$ in $B(x)$ unimodularni matriki in $\Pi(x)$ $k \times n$ diagonalna matrika z $r$ invariantnimi polinomskimi faktorji matrike $P(x)$ na diagonali in ničlami drugod:*

$$\Pi(x) = \begin{pmatrix} \gamma_1(x) & & & & & & & \\ & \gamma_2(x) & & & & & & \\ & & \ddots & & & & & \\ & & & \gamma_r(x) & & & & \\ & & & & 0 & & & \\ & & & & & \ddots & & \\ & & & & & & 0 \;\; \dots \;\; 0 \end{pmatrix}$$

*Matriki $\Pi(x)$ pravimo* **Smithova oblika** *matrike $P(x)$, elementom $\gamma_i(x)$ pa* **invariantni faktorji** *matrike $G(x)$.*

Tudi inverz matrike, ki jo lahko zapišemo v Smithovi obliki dobimo na preprost način. Ker sta $A(x)$ in $B(x)$ unimodularni, imata inverz. Potem lahko pišemo

$$P^{-1}(x) = B^{-1}(x)\Pi^{-1}(x)A^{-1}(x),$$

kjer je $\Pi^{-1}(x)$ matrika z diagonalnimi elementi $\gamma_i^{-1}(x)$. Torej je matrika $P^{-1}(x)$ polinomska takrat, ko so invariantni faktorji skalarji.

### 4.3.1 Smithov algoritem

V začetnem razdelku tega poglavja smo se opremili z orodji, ki nam omogočajo konstrukcijo algoritma, ki pripelje matriko iz njene osnovne oblike v Smithovo obliko. Ta algoritem bomo seveda imenovali *Smithov algoritem*. V grobem algoritem sestoji iz niza matričnih operacij, ki jih izvajamo na neki dani matriki. Najprej bomo opisali te operacije, potem bomo predstavili in dokazali Smithov algoritem ter na koncu izpeljali še nekaj lastnosti konvolucijskih kod, ki sledijo iz Smithove oblike matrike.

Smithov algoritem deluje tako, da s kombiniranjem dveh elementarnih operacij našo začetno matriko $P(x)$ pretvori v bločno matriko

$$P'(x) = \begin{pmatrix} \gamma(x) & 0 \\ 0 & P''(x) \end{pmatrix},$$

ki je ekvivalentna matriki $P(x)$ in kjer je $\gamma(x)$ polinom. Prvi invariantni faktor potem poiščemo iz prvega diagonalnega elementa matrike $P'(x)$. Za iskanje nadaljnih invariantnih faktorjev uporabimo Smithov algoritem na matriki $P''(x)$. Predpostavimo še lahko, da ima $P(x)$ vsaj en neničelen element, sicer nimamo kaj računati.

Vpeljimo še operaciji, o katerih je bilo govora. Ključnega pomena pri temu je, da obe operaciji lahko predstavimo v matrični obliki in tako operaciji na dani matriki izpeljemo tako, da začetno matriko množimo z matrično formo, ki predstavlja določeno operacijo.

Prva operacija je *zamenjava dveh vrstic ali stolpcev*. Predstavimo jo z (nesingularno) matriko

$$P_{ij} = \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & & & & & & & \\ & & & 0 & - & - & - & 1 & - & - & - \\ & & & \vert & 1 & & & \vert & & & \\ & & & \vert & & \ddots & & \vert & & & \\ & & & \vert & & & 1 & \vert & & & \\ & & & 1 & - & - & - & 0 & - & - & - \\ & & & & & & & & 1 & & \\ & & & & & & & & & \ddots & \\ & & & & & & & & & & 1 \end{pmatrix}$$

kjer sta enici izven diagonal v $i$-ti in $j$-ti vrstici. Za to matriko očitno velja $P_{ij}^{-1} = P_{ij}$. Prav tako velja $\det P_{ij} = -1 = 1$, ker računamo v $\mathbb{F}_2$.

Druga izmed imenovanih opercij je *dodajanje elementov ene vrstice (ali stolpca) ustreznim elementom druge vrstice (ali stolpca), pomnoženim z nekim polinomom iz* $\mathbb{F}_2[D]$.

$$R_{ij}(p(D)) = \begin{pmatrix} 1 & & & & & & & & & \\ & \ddots & & & & & & & & \\ & & 1 & - & - & - & p(D) & - & - & \\ & & & & & & \vert & & & \\ & & & & \ddots & & \vert & & & \\ & & & & & & \vert & & & \\ & & & & & & 1 & - & - & \\ & & & & & & & \ddots & & \\ & & & & & & & & 1 & \end{pmatrix}$$

Vidimo, da velja $R_{ij}^{-1}(p(D)) = R_{ij}(-p(D)) = R_{ij}(p(D))$ ter $\det R_{ij}(p(D)) = 1$.

Spomnimo se še, da matrično množenje z leve ustreza izvajanju operacij na vrsticah, medtem ko množenje z desne ustreza izvajanju operacij na stolpcih.

Zdaj lahko podamo formulacijo ključnega izreka tega poglavja.

**Izrek 4.3.3** (**Smithov izrek**). *Naj bo $G(D)$ binarna polinomska matrika velikosti $k \times n, k \leq n$, ranga $r$. Potem lahko matriko $G(D)$ zapišemo kot*

$$G(D) = A(D)\Gamma(D)B(D),$$

*kjer sta $A(D)$ in $B(D)$ binarni polinomski matriki zaporedoma velikosti $k \times k$ in $n \times n$ in katerih determinanta je enaka 1 ter kjer je $\Gamma(D)$ Smithova oblika matrike $G(D)$.*
*Elementi matrike $\Gamma(D)$ so neničelni elementi $\gamma_i(D) \in \mathbb{F}_2[D], 1 \leq i \leq r,$, so enolično določeni polinomi in zadoščajo relaciji*

$$\gamma_i(D) \mid \gamma_{i+1}(D),\ i = 1, \ldots, r-1.$$

*Če z $\Delta_i(D) \in \mathbb{F}_2[D]$ označimo največji skupni delitelj vseh $i \times i$ poddeterminant (minorjev) v $G(D)$, potem velja*

$$\gamma_i(D) = \frac{\Delta_i(D)}{\Delta_{i-1}(D)},$$

*kjer je $\Delta_0(D) = 1$ po definiciji in $i = 1, 2, \ldots, r$.*

**Dokaz.** Če je $G(D) = 0$, potem nimamo česa dokazovati. Zato predpostavimo, da je $G(D) \neq 0$.
Najprej moramo dokazati, da lahko s pomočjo elementarnih operacij (ki smo jih definirali pred izrekom) pridemo od matrike $G(D)$ do matrike, katere element v zgornjem levem kotu ni ničelen in ima najmanjšo stopnjo med vsemi neničelnimi elementi matrike, ki jih lahko pripeljemo v zgornji levi kot z omenjenima operacijama. Zdaj pa predpostavimo, da smo element z najmanjšo stopnjo že pripeljali v zgornji levi kot. Označimo elemente v tej novi matriki z $\alpha_{ij}(D), 1 \leq i \leq k, 1 \leq j \leq n$. Delimo nek element v prvi vrstici, $\alpha_{1j}(D), j > 1$, z $\alpha_{11}(D)$. Potem velja:

$$\alpha_{1j}(D) = \alpha_{11}(D)\beta_j(D) + \beta_{1j}(D),$$

kjer je $\deg(\beta_{1j}(D)) < \deg(\alpha_{11}(D))$. Zdaj prištejmo prvi stolpec, pomnožen z $\beta_j(D)$, k $j$-temu stolpcu. Ta operacija zamenja $\alpha_{1j}(D)$ z $\beta_{1j}(D)$. Torej, če $\beta_{1j}(D) \neq 0$, dobimo matriko, katere elementu z najnižjo stopnjo smo le-to pravkar zmanjšali. To proceduro ponovimo na novodobljeni matriki. Na podoben način ukrepamo tudi takrat, ko hočemo zmanjšati najnižjo stopnjo v prvemu stolpcu. Očitno se ta procedura enkrat konča, saj na vsakemu koraku stopnjo zmanjšamo (dejansko je opisani algoritem posplošitev Evklidovega algoritma za deljenje polinomov). Na koncu dobimo neko matriko $G'(D) = (\beta_{ij}(D))$, v kateri ima element $\beta_{11}(D)$ najmanjšo stopnjo in deli tako $\beta_{1j}(D)$ kot tudi $\beta_{i1}(D)$ za vse $i$ in $j$. Z uporabo prej definiranih operacij nato "počistimo" elemente v prvi vrstici in prvemu stolpcu (vse elemente razen $\beta_{11}(D)$). Tako dobimo matriko oblike

$$\begin{pmatrix} \beta_{11}(D) & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & G''(D) & \\ 0 & & & \end{pmatrix}$$

Označimo $\beta_{11}(D)$ z $\gamma_1(D)$.
Zdaj dokažimo, da deli $\gamma_1(D)$ vsak element v $G''(D) = (\delta_{ij}(D))$. Če to ne bi držalo, potem bi lahko s seštevanjem $j$-tega in prvega stolpca dobili nov prvi stolpec. S ponavljanjem tega postopka bi v zgornjem levem kotu dobili element z manjšo stopnjo kot jo ima $\beta_{11}(D)$. To pa je protislovje. Zato lahko matriko pišemo v obliki

$$\begin{pmatrix} \gamma_1(D) & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \gamma_1(D)G^*(D) & \\ 0 & & & \end{pmatrix}$$

Ponavljanje pravkar opisanega postopka na matriki $G^*(D)$ nam dá dokaz za prvi del izreka.
Dokažimo zdaj, da na

$$\Delta_i(D) = \text{največji skupni delitelj vseh } i \times i \text{ minorjev matrike } G(D)$$

ne vplivajo elementarne operacije na vrsticah in stolpcih matrike $G(D)$. Naj bo $A(D) = (a_{ij}(D))$ matrika velikosti $k \times k$ z elementi iz $\mathbb{F}_2[D]$, ki jo dobimo kot produkt elementarnih operacij na vrsticah matrike $A(D)$. Potem je element na mestu $(i, j)$ v matriki $A(D)G(D)$ enak

$$\sum_{\ell} a_{i\ell}(D) g_{\ell j}(D).$$

Vrstice matrike $A(D)G(D)$ so torej linearne kombinacije vrstic matrike $G(D)$. Zato so tudi $(i \times i)$-minorji matrike $A(D)G(D)$ linearne kombinacije $i \times i$ minorjev matrike $G(D)$. Od tod pa že sledi, da je največji skupni delitelj $i \times i$ minorjev matrike $G(D)$ delitelj največjega skupnega delitelja $i \times i$ minorjev matrike $A(D)G(D)$. Ker je $\det A(D) = 1$, je tudi $A^{-1}(D)$ polinomska matrika velikosti $k \times k$. S ponavljanjem zgornje razlage za matriko $A^{-1}(D)(A(D)G(D))$ lahko dokažemo, da je največji skupni delitelj $i \times i$ minorjev matrike $A(D)G(D)$ delitelj največjega skupnega delitelja $i \times i$ minorjev matrike $A^{-1}(D)(A(D)G(D)) = G(D)$. Od tod sklepamo, da sta največja skupna delitelja $i \times i$ minorjev matrik $G(D)$ in $A(D)G(D)$ enaka. Ker pa je

$A(D)$ lahko produkt katerihkoli elementarnih operacij na vrsticah, smo pokazali, da je $\Delta_i(D)$ neodvisen od teh operacij. Podobno pokažemo, da enako velja za operacije na stolpcih.

Iz pravkar dokazanega sledi identiteta

$$\Delta_i(D) = \text{največji skupni delitelj vseh } i \times i \text{ minorjev matrike } \Gamma(D).$$

Iz $\Gamma(D)$ pa lahko preberemo, da velja

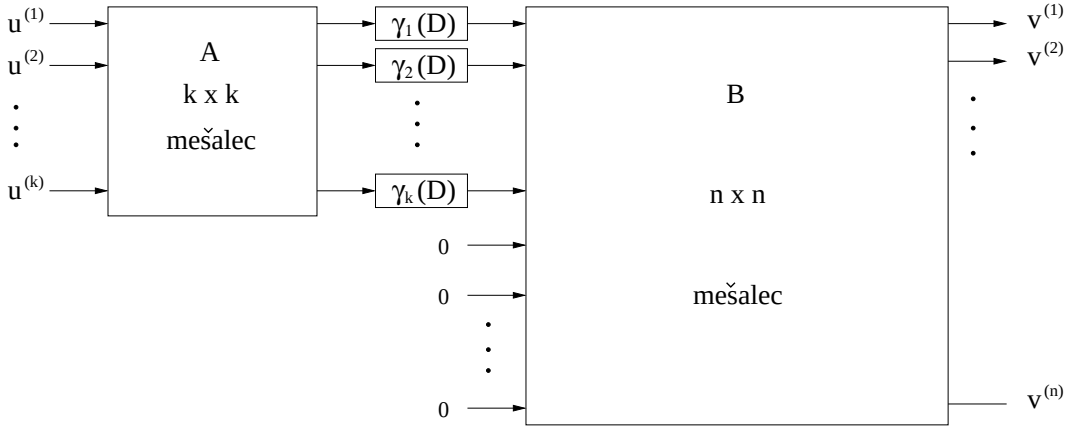
$$\begin{aligned}
\Delta_1(D) &= \gamma_1(D), \\
\Delta_2(D) &= \gamma_1(D)\gamma_2(D), \\
&\vdots \\
\Delta_r(D) &= \gamma_1(D)\gamma_2(D)\ldots\gamma_r(D),
\end{aligned}$$

oziroma, upoštevajoč $\Delta_0(D) = 1$, še

$$\gamma_i(D) = \frac{\Delta_i(D)}{\Delta_{i-1}(D)}, i = 1, 2, \ldots, r.$$

Ker pa enoličnost $\Delta_i(D), i = 1, 2, \ldots, r$, implicira enoličnost $\gamma_i(D), i = 1, 2, \ldots, r$, je dokaz končan. □

Poglejmo, kaj pomeni Smithova oblika za konvolucijsko kodo in njeno morebitno implementacijo. Matrikam $A(D)$ in $B(D)$ lahko rečemo kar "mešalci", saj opisujejo permutacije, ki jih naredimo na začetni generatorski matriki (ki je, mimogrede, polnega ranga). Vhodna zaporedja premešamo v $k \times k$ mešalcu. Kar dobimo na izhodu iz mešalcev pomnožimo z $k$ invariantnimi faktorji. Dobljeni produkt skupaj z $n - k$ ničlami pa ponovno premešamo v $n \times n$ mešalcu, da dobimo izhodna zaporedja. Vse pravkar opisano lepo ilustrira spodnja slika.



Slika 13 - Kodiranje konvolucijske kode z uporabo Smithovega izreka

Naslednja trditev je direktna posledica Smithovega algoritma.

**Posledica 4.3.4.** *Naj bo $k \leq n$. Vsako binarno polinomsko matriko $T(x)$ velikosti $k \times n$ lahko zapišemo v Smithovi obliki.* □

## 4.4 Ekvivalentne in bazične kodirne matrike

V komunikacijskem kontekstu je pomembno vedeti, kdaj dve matriki določata enako kodo, saj nam to omogoča sestavo najmanj kompleksnih kodirnikov. Zato vpeljemo naslednjo definicijo ekvivalence kodirnih matrik in kodirnikov.

**Definicija 4.4.1.** *Kodirni konvolucijski generatorski matriki $G(D)$ in $G'(D)$ sta* **ekvivalentni***, če je rezultat njunega kodiranja enaka koda. Konvolucijska kodirnika sta* **ekvivalentna***, če sta njuni generatorski matriki ekvivalentni.*

Velja naslednja trditev.

**Izrek 4.4.2.** *Konvolucijski generatorski matriki $G(D)$ in $G'(D)$ z informacijsko stopnjo $R = k/n$ sta ekvivalentni natanko takrat, ko obstaja taka nesingularna matrika $T(D)$ reda $k \times k$ nad $\mathbb{F}_2(D)$, da je*

$$G(D) = T(D)G'(D).$$

**Dokaz.**
($\Rightarrow$) Če velja $G(D) = T(D)G'(D)$, sta matriki $G(D)$ in $G'(D)$ ekvivalentni.

($\Leftarrow$) Predpostavimo, da sta $G(D)$ in $G'(D)$ ekvivalentni. Naj bo $\boldsymbol{g}_i(D) \in \mathbb{F}_2^n(D)$ $i$-ta vrstica matrike $G(D)$. Potem obstaja tak $\boldsymbol{u}_i(D) \in \mathbb{F}_2^k(D)$, da je

$$\boldsymbol{g}_i(D) = \boldsymbol{u}_i(D)G'(D).$$

Naj bo zdaj

$$\begin{pmatrix} \boldsymbol{u}_1(D) \\ \boldsymbol{u}_2(D) \\ \vdots \\ \boldsymbol{u}_k(D) \end{pmatrix}$$

Potem je

$$G(D) = T(D)G'(D),$$

kjer je $T(D)$ matrika velikosti $k \times k$ nad $\mathbb{F}_2(D)$. Označimo z $S'(D)$ $k \times k$ nesingularno podmatriko matrike $G'(D)$ in z $S(D)$ ustrezno $k \times k$ podmatriko matrike $G(D)$. Potem je $S(D) = T(D)S'(D)$. Torej je $T(D) = S(D)S'(D)^{-1}$ in zato je $T(D)$ matrika nad $\mathbb{F}_2(D)$. Ker ima $G(D)$ kot generatorska matrika rang $k$, je tudi $T(D)$ ranga $k$ in zato nesingularna. □

**Definicija 4.4.3.** *Konvolucijska generatorska (kodirna) matrika je* **bazična**, *če je polinomska in ima polinomski desni inverz. Konvolucijski kodirnik je* **bazičen**, *če je njegova generatorska matrika bazična.*

Za nadaljnjo obravnavo bazičnih matrik bomo potrebovali dve pomožni trditvi, ki nam bosta pomagali izpeljati karakterizacijo bazičnih generatorskih matrik.

**Trditev 4.4.4.** *Konvolucijska generatorska matrika $G(D)$ z izvedljivim desnim inverzom je kodirna matrika.*

**Dokaz.** Naj bo $G^{-1}(D)$ izvedljiv desni inverz matrike $G(D)$. Torej velja

$$G(D)G^{-1}(D) = I_k.$$

Če vstavimo 0 namesto $D$, dobimo

$$G(0)G^{-1}(0) = I_k,$$

torej ima $G(0)$ poln rang. □

**Posledica 4.4.5.** *Bazična generatorska matrika je bazična kodirna matrika.*

Zdaj smo pripravljeni na osrednji izrek tega razdelka.

**Izrek 4.4.6.** *Vsaka racionalna generatorska matrika je ekvivalentna neki bazični kodirni matriki.*

**Dokaz.** Iz izreka 4.2.6 vemo, da ima vsaka racionalna generatorska matrika ekvivalentno polinomsko generatorsko matriko, ki je brez zamika. Označimo to ekvivalentno matriko z $G(D)$. Kot vemo iz Smithovega izreka, $G(D)$ lahko zapišemo v obliki $G(D) = A(D)\Gamma(D)B(D)$, kjer sta $A(D)$ in $B(D)$ zaporedoma $k \times k$ in $n \times n$ polinomski matriki z enotsko determinanto in je

$$\Gamma(D) = \begin{pmatrix} \gamma_1(D) & & & & & \\ & \gamma_2(D) & & & & \\ & & \ddots & & & \\ & & & \gamma_k(D) & 0 & \dots & 0 \end{pmatrix}$$

Naj bo $G'(D)$ generatorska matrika, sestavljena iz prvih $k$ vrstic matrike $B(D)$. Potem je

$$\Gamma(D) = A(D) \begin{pmatrix} \gamma_1(D) & & & \\ & \gamma_2(D) & & \\ & & \ddots & \\ & & & \gamma_k(D) \end{pmatrix} B(D)$$

Ker sta tako $A(D)$ kot tudi

$$\begin{pmatrix} \gamma_1(D) & & & \\ & \gamma_2(D) & & \\ & & \ddots & \\ & & & \gamma_k(D) \end{pmatrix}$$

nesingularni matriki nad $\mathbb{F}_2(D)$, iz izreka 4.4.2 sledi, da sta $G(D)$ in $G'(D)$ ekvivalentni. Ker pa vemo tudi, da je $G'(D)$ polinomska in ker ima $B(D)$ polinomski inverz, ima tudi $G'(D)$ polinomski desni inverz (sestavljen iz prvih $k$ stolpcev matrike $B^{-1}(D)$). Zato je $G'(D)$ bazična generatorska matrika. Zadnji izrek nam pa pove, da je potem $G'(D)$ bazična kodirna matrika. □

Vsaka polinomska konvolucijska generatorska matrika ima polinomski desni inverz brez zamika natanko takrat, ko v Smithovi formi velja $\gamma_k(D) = 1$. Zato velja naslednja trditev.

**Izrek 4.4.7.** *Generatorska matrika je bazična natanko takrat, ko je polinomska in je* $\gamma_k(D) = 1$.

Ta izrek nam dá karakterizacijo bazičnih generatorskih matrik. S pomočjo Smithovega izreka zdaj lahko strnemo dosedanje rezultate tega razdelka v sledečo trditev.

**Izrek 4.4.8.** *Bazično kodirno matriko* $G(D)$ *lahko zapišemo v Smithovi obliki kot*

$$G(D) = A(D)\Gamma(D)B(D),$$

*kjer je* $A(D)$ $k \times k$ *polinomska matrika z enotsko determinanto,* $B(D)$ $n \times n$ *polinomska matrika z enotsko determinanto in* $\Gamma(D)$ $k \times n$ *matrike oblike*

$$\Gamma(D) = \begin{pmatrix} 1 & & & & & \\ & 1 & & & & \\ & & \ddots & & & \\ & & & 1 & 0 & \dots & 0 \end{pmatrix}$$

□

Od tod pa sledi še naslednji rezultat.

**Posledica 4.4.9.** *Bazična kodirna matrika je nekatastrofična.* □

Oglejmo si še, kako je z ekvivalenco bazičnih kodirnih matrik. Dejansko smo tako orodja kot rezultate že pripravili, zato nam preostane samo njihova formulacija.

**Izrek 4.4.10.** *Bazični kodirni matriki $G(D)$ in $G'(D)$ sta ekvivalentni natanko takrat, ko je $G'(D) = T(D)G(D)$, kjer je $T(D)$ polinomska matrika velikosti $k \times k$ z enotsko determinanto.*

**Dokaz.**
($\Leftarrow$) Naj bo $G'(D) = T(D)G(D)$, kjer je $T(D)$ polinomska matrika velikosti $k \times k$ z enotsko determinanto. Po izreku 4.4.2 sta matriki $G(D)$ in $G'(D)$ ekvivalentni.
($\Rightarrow$) Predpostavimo, da sta $G(D)$ in $G'(D)$ ekvivalentni. Po izreku 4.4.2 obstaja taka nesingularna $k \times k$ matrika $T(D)$ nad $\mathbb{F}_2(D)$, da je $G'(D) = T(D)G(D)$. Matrika $G(D)$ je bazična, zato ima polinomski desni inverz $G^{-1}(D)$. Zato je tudi matrika $T(D) = G'(D)G^{-1}(D)$ polinomska. Enako sklepamo, če zamenjamo vlogi $G(D)$ in $G'(D)$. Na ta način dobimo $G(D) = S(D)G'(D)$ za neko polinomsko matriko $S(D)$. Od tod pa sledi, da je $G(D) = S(D)T(D)G(D)$. Ker ima $G(D)$ poln rang, lahko sklepamo, da velja enakost $S(D)T(D) = I_k$. Ker pa sta $S(D)$ in $T(D)$ polinomski, morata imeti enotsko determinanto. S tem je dokaz končan. □

Za konec predstavimo še eno uporabo pravkar dokazanega izreka.

**Posledica 4.4.11.** *Naj bo $G(D) = A(D)\Gamma(D)B(D)$ Smithova dekompozicija bazične kodirne matrike $G(D)$ in naj bo $G'(D)$ $k \times n$ polinomska matrika, sestavljena iz prvih $k$ vrstic matrike $B(D)$. Tedaj sta $G(D)$ in $G'(D)$ ekvivalentni bazični kodirni matriki.*

**Dokaz.** Ker je $G(D)$ bazična, iz izreka 4.4.8 sledi, da je $G(D) = A(D)G'(D)$, kjer je $A(D)$ $k \times k$ unimodularna matrika. Iz izreka 4.4.10 pa sledi nato željeni rezultat.□

## 4.5 Minimalno-bazične in minimalne kodirne matrike in kodirniki

### 4.5.1 Minimalno-bazičnost

V tem razdelku bomo iz različnih matričnih lastnosti izpeljali kar nekaj koristnih napotkov za implementacijo kodirnikov za praktične namene. Ključni pojem razdelka je minimalni kodirnik, torej tak kodirnik, ki rabi kar najmanj spomina za kodiranje konvolucijske kode za dano generatorsko matriko.

Na začetku bomo vpeljali nekaj pojmov, s katerimi si bomo pomagali pri izpeljavi zgoraj omenjenih lastnosti. Začnimo z nekaj definicijami. *Možna dolžina i-tega vhoda* za polinomsko generatorsko matriko $G(D)$je enaka

$$\nu_i = \max_{1 \le j \le n} \{\deg g_{ij}(D)\}.$$

*Spomin* polinomske generatorske matrike je maksimum možnih dolžin:

$$m = \max_{1 \le i \le k} \{\nu_i\}.$$

*Skupna možna dolžina* predstavlja vsoto možnih dolžin

$$\nu = \sum_{i=1}^{k} \nu_i.$$

Polinomsko generatorsko matriko torej lahko implementiramo s pomočjo linearnega zaporednega registra, ki sestoji iz $k$ pomičnih registrov, kjer ima $i$-ti register dolžino $\nu_i$, in katerega izhode računamo kot vsote ustreznih vsebin pomičnih registrov po modulu 2.

Zdaj bomo izpeljali karakterizacijo bazičnih kodirnih matrik, za katere potrebujemo najmanj spomina za dani razred ekvivalentnih bazičnih kodirnih matrik. Definirajmo najprej ključni pojem tega razdelka.

**Definicija 4.5.1. Minimalno-bazična** *kodirna matrika je taka bazična kodirna matrika, katere skupna dolžina $\nu$ je najmanjša za vse ekvivalentne bazične kodirne matrike.*

Kasneje bomo videli, da so minimalno-bazične matrike minimalne tudi v bolj splošnem smislu.

Označimo z $G(D)$ bazično kodirno matriko. Pomembno vlogo v prihodnjih izrekih bodo igrali tudi koeficienti polinomov najvišje stopnje, ki jih bomo gledali po vrsticah. Zato označimo z $[G(D)]_h$ matriko, ki ima element 1 na mestu $(i, j)$, za katerega je $\deg g_{ij}(D) = \nu_i$ in 0 sicer.

Naslednji izrek predstavlja karakterizacijo minimalno-bazičnih kodirnih matrik.

**Izrek 4.5.2.** *Naj bo $G(D)$ bazična kodirna matrika velikosti $k \times n$ s skupno dolžino $\nu$. Naslednje trditve so ekvivalentne:*

*(i) $G(D)$ je minimalno-bazična kodirna matrika.*

*(ii) Maksimalna stopnja $\mu$ vseh $k \times k$ poddeterminant za $G(D)$ je enaka skupni dolžini $\nu$.*

*(iii) Matrika $[G(D)]_h$ je polnega ranga.*

**Dokaz.** Pišimo

$$G(D) = G_0(D) + G_1(D), \tag{4.1}$$

kjer je

$$G_1(D) = \begin{pmatrix} D^{\nu_1} & & & \\ & D^{\nu_2} & & \\ & & \ddots & \\ & & & D^{\nu_k} \end{pmatrix} [G(D)]_h$$

Vsi elementi v $i$-ti vrstici matrike $G_0(D)$ imajo stopnjo, manjšo od $\nu_i$. Največja stopnja $\mu$ $k \times k$ poddeterminant matrike $G(D)$ je manjša ali enaka $\nu$. Že iz same oblike matrike $G_1(D)$ vidimo, da sta trditvi (ii) in (iii) ekvivalentni. Ostane torej samo dokaz ekvivalence med (i) in (ii).

(i) $\Rightarrow$ (ii): Predpostavimo, da je $G(D)$ minimalno-bazična. Predpostavimo prav tako, da je $\mu < \nu$, kar pomeni da ima $[G(D)]_h$ rang, manjši od $k$. Označimo z $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_k$ vrstice matrike $G(D)$ in z $[\boldsymbol{r}_1], [\boldsymbol{r}_2], \ldots, [\boldsymbol{r}_k]$ vrstice matrike $[G(D)]_h$. Potem velja naslednja enakost:

$$[\boldsymbol{r}_{i_1}] + [\boldsymbol{r}_{i_2}] + \ldots + [\boldsymbol{r}_{i_d}] = 0 \;\text{ za } i_j \in 1, \ldots, k.$$

$i$-ta vrstica matrike $G_1(D)$ je enaka $D^{\nu_i}[\mathbf{r}_i]$. Brez izgube splošnosti lahko predpostavimo, da je $\nu_{i_d} \geq \nu_{i_j}, j = 1, 2, \ldots, d-1$. Če izraz

$$D^{\nu_{i_d} - \nu_{i_1}} D^{\nu_{i_1}}[\boldsymbol{r}_{i_1}] + D^{\nu_{i_d} - \nu_{i_2}} D^{\nu_{i_2}}[\boldsymbol{r}_{i_2}] + \ldots + D^{\nu_{i_d} - \nu_{i_{d-1}}} D^{\nu_{i_{d-1}}}[\boldsymbol{r}_{i_d}] =$$

$$D^{\nu_{i_d}}([\boldsymbol{r}_{i_1}] + [\boldsymbol{r}_{i_2}] + \ldots + [\boldsymbol{r}_{i_{d-1}}])$$

prištejemo $i_d$-ti vrstici matrike $G_1(D)$, dobimo ničelno vrstico. Podobno, če izraz

$$\boldsymbol{r}(D) = D^{\nu_{i_d} - \nu_{i_1}} \boldsymbol{r}_1 + D^{\nu_{i_d} - \nu_{i_2}} \boldsymbol{r}_2 + \ldots + D^{\nu_{i_d} - \nu_{i_{d-1}}} \boldsymbol{r}_{i_{d-1}}$$

prištejemo $i_d$-ti vrstici matrike $G(D)$, zmanjšamo najvišjo stopnjo v $i_d$-ti vrstici matrike $G(D)$, medtem ko ostale vrstice v $G(D)$ ostanejo nespremenjene. Tako torej dobimo bazično kodirno matriko, ki je ekvivalentna $G(D)$ z manjšo skupno stopnjo, kot jo ima $G(D)$. To je v protislovju s predpostavko, da je $G(D)$ minimalno-bazična in zato lahko sklepamo, da je $\mu = \nu$.

(ii) $\Rightarrow$ (i): Naj bo $G'(D)$ bazična kodirna matrika, ki je ekvivalentna $G(D)$. Iz izreka 4.4.10 sledi, da je $G'(D) = T(D)G(D)$, kjer je $T(D)$ $k \times k$ polinomska matrika z enotsko determinanto. Ker je $\det T(D) = 1$, je najvišja stopnja $k \times k$ poddeterminant matrike $G'(D)$ enaka najvišji stopnji za matriko $G(D)$. To pomeni, da je $\mu$

invarianten za vse ekvivalentne bazične kodirne matrike. Ker pa je $\mu$ manjši ali enak skupni dolžini vseh ekvivalentnih bazičnih kodirnih matrik, je tudi $G(D)$ minimalno-bazična kodirna matrika. □

Iz prejšnjega izreka in dejstva, da je $\mu$ invarianten za vse ekvivalentne bazične kodirne matrike, sledi naslednji rezultat.

**Izrek 4.5.3.** *Naj bo $G(D)$ $k \times n$ bazična kodirna matrika z najvišjo stopnjo $k \times k$ poddeterminant $\mu$. Potem ima $G(D)$ ekvivalentno minimalno-bazično kodirno matriko s skupno dolžino $\mu$.* □

Za izračun minimalno-bazičnih matrik iz bazičnih kodirnih matrik obstaja algoritem, ki ga lahko najdemo v [2] in ga tukaj ne bomo opisali, je pa uporaben tudi pri izpeljavi rezultatov, ki sledijo in jih podajamo brez dokaza.

**Izrek 4.5.4.** *Vsaka racionalna generatorska matrika je ekvivalentna neki minimalno-bazični kodirni matriki.* □

**Izrek 4.5.5.** *Možne dolžine dveh ekvivalentnih minimalno-bazičnih kodirnih matrik so enake do vrstnega reda natančno.* □

**Izrek 4.5.6.** *Ekvivalentni minimalno-bazični kodirni matriki imata enak spomin.* □

### 4.5.2 Minimalnost

Kot je bilo že obljubljeno, bomo v tem razdelku pokazali, da so minimalno-bazične matrike minimalne tudi v bolj splošnem smislu. Za izpeljave, ki sledijo, bomo potrebovali nekaj začetnih definicij. Pomični register ali kakršnokoli realizacijo neke generatorske matrike bomo v tem razdelku imenovali *izvedba*.

**Definicija 4.5.7. Stanje kodirnika** *$\sigma$ neke izvedbe racionalne generatorske matrike $G(D)$ je vsebina njegovih spominskih elementov (oz. registrov). Množico stanj kodirnika imenujemo* **prostor stanj kodirnika**.

Če je $G(D)$ polinomska matrika, je dimenzija prostora stanj kodirnika enaka skupni možni dolžini $\nu$ matrike $G(D)$.

**Definicija 4.5.8.** *Naj bo $G(D)$ racionalna generatorska matrika.* **Abstraktno stanje $\boldsymbol{s}(D)$**, *ki ustreza vhodnemu zaporedju $\boldsymbol{u}(D)$, je zaporedje izhodov v času 0 in kasneje, ki jih dobimo iz tistega dela $\boldsymbol{u}(D)$, ki se pojavi do časa $-1$ in mu nato sledijo ničle. Množici vseh abstraktnih stanj pravimo* **prostor abstraktnih stanj**.

Poudarimo, da je abstraktno stanje odvisno le od generatorske matrike in na noben način od njene izvedbe. Število stanj kodirnika je večje ali enako od števila abstraktnih stanj. Različna abstraktna stanja morajo izhajati iz različnih stanj kodirnika v času 0.

Označimo s $P$ projekcijo, ki zaporedja odreže tako, da se končajo v času $-1$ in s $Q = 1 - P$ projekcijo, ki odreže zaporedja tako, da se začnejo v času 0. Za zaporedje

$$\boldsymbol{u}(D) = \boldsymbol{u}_d D^d + \boldsymbol{u}_{d+1} D^{d+1} + \ldots$$

torej velja

$$\boldsymbol{u}(D)P = \begin{cases} \boldsymbol{u}_d D^d + \boldsymbol{u}_{d+1} D^{d+1} + \ldots + \boldsymbol{u}_{-1} D^{-1}, & d < 0 \\ \boldsymbol{0}, & d \geq 0 \end{cases}$$

in

$$\boldsymbol{u}(D)Q = \boldsymbol{u}_0 + \boldsymbol{u}_1 D + \boldsymbol{u}_2 D^2 + \ldots$$

Iz definicije projekcij je očitno, da velja $P + Q = 1$. Torej lahko abstraktno stanje $\boldsymbol{s}(D)$, ki ustreza stanju kodirnika $\boldsymbol{u}(D)$ zapišemo kot

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)PG(D)Q.$$

Prostor stanj kodirnika za generatorsko matriko s skupno možno dolžino $\nu$ ima $2^\nu$ stanj.

Število abstraktnih stanj bo imelo ključno vlogo pri karakterizaciji minimalnih kodirnikov in ustreznih generatorskih matrik. Podajmo najprej lemo, ki jo bomo nato uporabili pri izpeljavi števila abstraktnih stanj za minimalno-bazične kodirne matrike.

**Lema 4.5.9.** *Naj bo $G(D)$ minimalno-bazična kodirna matrika in naj bo*

$$\boldsymbol{u}(D) = \sum_{i=-m}^{n} (u_i^{(1)},\ u_i^{(2)},\ \ldots,\ u_i^{(k)}) D^i,$$

*kjer je $m$ spomin matrike $G(D)$ in $n \geq -m$. Če je $\boldsymbol{u}(D)G(D)Q = 0$, potem je $\boldsymbol{u}(D) = 0$.*

**Dokaz.** Naj bo

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G(D).$$

Po predpostavki leme je

$$\boldsymbol{v}(D)Q = \boldsymbol{u}(D)G(D)Q = \boldsymbol{0}.$$

Torej mora biti vsak koeficient za $D^i, i \geq 0$, v vrsti $\boldsymbol{v}(D)$ enak $\boldsymbol{0}$. Če pišemo matriko $G(D)$ v obliki (4.1), imamo

$$\boldsymbol{v}(D) = \boldsymbol{u}(D)G_0(D) + \boldsymbol{u}(D) \begin{pmatrix} D^{\nu_1} & & & \\ & D^{\nu_2} & & \\ & & \ddots & \\ & & & D^{\nu_k} \end{pmatrix} [G(D)]_h$$

Brez izgube splošnosti lahko predpostavimo, da je

$$m = \nu_1 = \nu_2 = \ldots = \nu_\ell > \nu_{\ell+1} \geq \ldots \geq \nu_k.$$

Tedaj je koeficient pri $D^{m+n}, m+n \geq 0$, v $\boldsymbol{v}(D)$ enak

$$(u_n^{(1)},\ u_n^{(2)},\ ,\ldots,\ u_n^{(\ell)},\ 0,\ \ldots,\ 0)[G(D)]_h.$$

Ta izraz pa mora biti po predpostavki $\boldsymbol{u}(D)G(D)Q = \boldsymbol{0}$ enak $\boldsymbol{0}$. Ker je namreč $G(D)$ minimalno-bazična, ima $[G(D)]_h$ rang $k$. Zato velja $u_n^{(1)} = u_n^{(2)} = \ldots = u_n^{(\ell)} = 0$. Z nadaljevanjem tega sklepa pa pridemo do $\boldsymbol{u}(D) = \boldsymbol{0}$. □

Naslednji izrek nam bo karakteriziral minimalne izvedbe (torej, izvedbe z najmanj spominskimi elementi) za dano minimalno-bazično kodirno matriko.

**Izrek 4.5.10.** *Naj bo $G(D)$ minimalno-bazična kodirna matrika, katere skupna možna dolžina je enaka $\nu$. Tedaj je*

$$\sharp\{\textit{abstraktna stanja}\} = 2^\nu.$$

*( $\sharp$ označuje kardinalnost dane množice.)*

**Dokaz.** Vhodna zaporedja oblike

$$\boldsymbol{u}(D) = \left( \sum_{i=1}^{\nu_1} u_{-i}^{(1)} D^{-i},\ \ \sum_{i=1}^{\nu_2} u_{-i}^{(2)} D^{-i},\ \ \ldots,\ \ \sum_{i=1}^{\nu_k} u_{-i}^{(k)} D^{-i} \right)$$

nam dajo vsa možna stanja kodirnika v času 0. Torej imamo abstraktna stanja

$$\boldsymbol{s}(D) = \boldsymbol{u}(D)G(D)Q.$$

Vsako abstraktno stanje lahko dobimo na ta način. Torej je

$$\sharp\{\textit{abstraktna stanja}\} \leq 2^\nu.$$

Za dokaz enakosti je zaradi linearnosti dovolj pokazati, da je $\boldsymbol{u}(D) = \boldsymbol{0}$ edini vhod, ki proizvede abstraktno stanje $\boldsymbol{s}(D) = \boldsymbol{0}$. To dejstvo pa sledi iz prejšnje leme. □

Zdaj lahko vpeljemo minimalnost.

**Definicija 4.5.11.** *Konvolucijska generatorska matrika je* **minimalna**, *če je število njenih abstraktnih stanj najmanjše med vsemi njej ekvivalentnimi generatorskimi matrikami.*

Za karakterizacijo minimalnih generatorskih matrik bomo potrebovali še nekaj pomožnih trditev.

**Lema 4.5.12.** *Samo ničelno abstraktno stanje minimalno-bazične kodirne matrike $G(D)$ je lahko kodna beseda.* □

**Lema 4.5.13.** *Naj bo $G(D)$ generatorska matrika ekvivalenta minimalno-bazični kodirni matriki $G_{mb}$. Potem velja:*

$$\sharp\{\textit{abstraktna stanja za } G(D)\} \geq \sharp\{\textit{abstraktna stanja za } G_{mb}(D)\}.\square$$

Iz obeh pravkar podanih trditev sledi naslednja trditev.

**Trditev 4.5.14.** *Vsaka minimalno-bazična kodirna matrika je minimalna kodirna matrika.* □

Vzemimo neko Laurentovo vrsto $f(D)$. *Razpon* vrste $f(D)$ je interval, določen z indeksi prve in zadnje neničelne komponente v $f(D)$, če slednja obstaja. Če zadnja neničelna komponenta ne obstaja, je razpon neskončen. Torej je

$$\text{razpon } f(D) = \begin{cases} [\text{del } f(D), \deg f(D)], & \text{če } \deg f(D) < \infty \\ [\text{del } f(D), \infty), & \text{sicer} \end{cases}$$

("del" označuje prvo nenegativno komponento, "deg" pa stopnjo $f(D)$.)
Vse ugotovitve tega podrazdelka povzamemo v naslednjem izreku.

**Izrek 4.5.15.** *Naj bo $G(D)$ generatorska matrika in $G_{mb}(D)$ njej ekvivalentna minimalno-bazična kodirna matrika. Naslednje trditve so ekvivalentne:*

*(i) $G(D)$ je minimalna generatorska matrika.*

*(ii) $\sharp\{\textit{abstraktna stanja za } G(D)\} = \sharp\{\textit{abstraktna stanja za } G_{mb}(D)\}$*

*(iii) Samo ničelno abstraktno stanje za $G(D)$ je lahko kodna beseda.*

*(iv) Matrika $G(D)$ ima polinomski desni inverz spremenljivke $D$ in polinomski desni inverz spremenljivke $D^{-1}$.*

*(v)* razpon $\boldsymbol{u}(D) \subseteq$ razpon $\boldsymbol{u}(D)G(D)$ *za vsa racionalna vhodna zaporedja $\boldsymbol{u}(D)$.*
□

Dokaz tega izreka bomo izpustili, saj smo večino trditev v izreku omenili v pripravah na formulacijo izreka. Iz tega izreka in izreka 4.4.4 pa očitno sledijo še naslednje trditve.

**Posledica 4.5.16.** *Minimalna generatorska matrika je minimalna kodirna matrika.*

**Posledica 4.5.17.** *Minimalna kodirna matrika je nekatastrofična.*

Za konec razdelka podajmo še definicijo minimalnega kodirnika, ki predstavlja izvedbo minimalne kodirne matrike.

**Definicija 4.5.18. Minimalni kodirnik** *je izvedba minimalne kodirne matrike $G(D)$ z minimalnim številom spominskih elementov za vse možne izvedbe $G(D)$.*

## 4.6 Sistematični konvolucijski kodirniki

Med konvolucijskimi kodami velja omeniti tudi kode, pri katerih lahko iz kodne besede preberemo vhodno informacijo.

**Definicija 4.6.1.** *Konvolucijskemu kodirniku z informacijsko stopnjo $R = k/n$, pri kateremu $k$ vhodnih informacijskih zaporedij nastopa nespremenjeno v $n$ kodnih zaporedjih pravimo* **sistematični kodirnik**, *njegovi generatorski matriki pa* **sistematična generatorska matrika**.

Če konvolucijsko kodo kodiramo s pomočjo sistematične generatorske matrike, potem lahko vedno permutiramo njene stolpce tako, da $k$ (nespremenjenih) vhodnih informacijskih zaporedij nastopi na začetku $n$ kodnih zaporedij. Ker vemo, da nam permutacije stolpcev začetne generatorske matrike dajo ekvivalentno generatorsko matriko in zato tudi isto konvolucijsko kodo, lahko brez izgube splošnosti pišemo sistematične generatorske matrike v obliki

$$G(D) = (I_k \; R(D)),$$

kjer sta $I_k$ enotska matrika reda $k$ in $R(D)$ matrika velikosti $k \times (n-k)$, katere elementi so racionalne funkcije v $D$. Dodajmo še, da je sistematičnost lastnost generatorske matrike in ne konvolucijske kode. To sledi iz dejstva, da koda ni odvisna od konkretne konvolucijske preslikave. Koda je namreč množica kodnih zaporedij, ki izhajajo iz množice vhodnih zaporedij. Zato ima vsaka konvolucijska koda tako sistematične kot nesistematične generatorske matrike.

Sistematične konvolucijske generatorske matrike so ponavadi lažje za implementacijo. Njihovi desni inverzi so trivialni. Dokler se omejujemo na polinomske generatorske matrike (drugače rečeno, ne uporabljamo povratnih pomičnih registrov) pa so manj močne, kadar jih uporabljamo v kombinaciji z odkodiranjem na principu najverjetnejšega kandidata.

Ker vsebuje sistematična generatorska matrika enotsko podmatriko reda $k \times k$, je očitno, da velja naslednji rezultat.

**Izrek 4.6.2.** *Vsaka sistematična generatorska matrika je sistematična kodirna matrika.* □

Prav tako vemo, da je v vsaki bazični kodirni matriki največji skupni delitelj vseh $k \times k$ minorjev enak 1. Zato mora obstajati $k \times k$ podmatrika, katere determinanta je polinom brez zamika, saj bi bile sicer vse poddeterminante deljive z $D$. Če pa matriko z leve pomnožimo z inverzom prej omenjene podmatrike, dobimo ekvivalentno sistematično kodirno matriko, ki je morda racionalna. Zgornja razlaga predstavlja skico dokaza naslednjega izreka.

**Izrek 4.6.3.** *Vsaka konvolucijska generatorska matrika je ekvivalentna neki sistematični racionalni kodirni matriki.* □

Pred zaključkom tega razdelka pa povejmo še kaj o minimalnosti sistematičnih kodirnih matrik. Že prej smo videli, da ima vsaka sistematična kodirna matrika obliko

$$G(D) = (I_k \; R(D)),$$

kjer je $I_k$ enotska matrika reda $k$, $R(D)$ pa $k \times (n-k)$ matrika, katere elementi so racionalne funkcije spremenljivke $D$. Desni inverz matrike $G(D)$ je $n \times k$ matrika oblike

$$G^{-1}(D) = \begin{pmatrix} I_k \\ 0 \end{pmatrix}$$

ki je očitno polinom v $D$ in $D^{-1}$. Zato velja tudi izrek, s katerim končujemo razdelek o sistematičnih generatorskih matrikah.

**Izrek 4.6.4.** *Vsaka sistematična kodirna matrika je minimalna.* □

## 4.7 Kanonične generatorske matrike

Kot smo v predhodnih razdelkih videli, lahko konvolucijsko kodo definiramo s pomočjo različnih vrst generatorskih matrik, vključno s tistimi, katerih elementi so racionalne funkcije spremenljivke $D$. Naša pozornost je bila usmerjena predvsem v polinomske matrike. V tem razdelku bomo izpostavili še eno prav posebno vrsto generatorskih matrik, ki so zelo zanimive za implementacijo. Imenujemo jih *kanonične generatorske matrike* konvolucijske kode.

Preden podamo formalno definicijo kanoničnih generatorskih matrik, vpeljimo nekaj pojmov, ki nam bodo v pomoč v nadaljevanju. Označimo z $G(D)$ $k \times n$ polinomsko matriko z $(i,j)$-tim elementom $g_{ij}(D)$. Maksimalni stopnji elementov v $i$-ti vrstici recimo *stopnja $i$-te vrstice* matrike $G(D)$. Dodatno imenujmo vsoto stopenj vrstic matrike $G(D)$ *zunanja stopnja* matrike $G(D)$ in jo označimo z extdeg $G(D)$.

**Definicija 4.7.1. Kanonična generatorska matrika** *konvolucijske kode $C$ je polinomska generatorska matrika, ki ima med vsemi polinomskimi generatorskimi matrikami konvolucijske kode $C$ najmanjšo zunanjo stopnjo.*

Vsaka konvolucijska koda ima kanonično generatorsko matriko. Stopnji kanonične generatorske matrike rečemo *stopnja kode $C$*.
Imejmo $k \times n$ polinomsko matriko $G(D)$, kjer je $k \leq n$. Vseh $k \times k$ minorjev matrike $G(D)$ je $\binom{n}{k}$. *Notranja stopnja* matrike $G(D)$ z oznako intdeg $G(D)$ je maksimalna stopnja med vsemi $k \times k$ minorji matrike $G(D)$.

**Definicija 4.7.2. Bazična** *generatorska polinomska matrika konvolucijske kode je katerakoli polinomska generatorska matrika z najmanjšo notranjo stopnjo.*

Podajmo še eno definicijo, ki jo bomo uporabljali pri dokazovanju izrekov tega razdelka.

**Definicija 4.7.3.** *Polinomska matrika $G(D)$ je* **reducirana***, če ima $G(D)$ najmanjšo zunanjo stopnjo med vsemi polinomskimi matrikami oblike $UG$, kjer je $U$ $k \times k$ unimodularna matrika.*

Ker lahko vsako unimodularno matriko predstavimo kot produkt elementarnih matrik, je matrika reducirana če in samo če njene zunanje stopnje ne moremo zmanjšati z zaporedjem elementarnih operacij na vrsticah dane matrike. Obstaja nekaj ekvivalentnih formulacij bazičnih in reduciranih matrik. Spodaj jih bomo podali brez dokaza. Omenimo le, da so naštete ekvivalence samo nekatere izmed doslej znanih. Zato bomo omenili le najbolj nazorne.

**Izrek 4.7.4.** *Naj bo $G(D)$ polinomska generatorska matrika $(n, k)$ konvolucijske kode. Matrika $G(D)$ je bazična če in samo če velja katerikoli od sledečih trditev.*

*(i) Največji skupni delitelj $k \times k$ minorjev matrike $G(D)$ je 1.*

*(ii) Obstaja taka polinomska $n \times k$ matrika $K$, da je $GK = I_k$.*

*(iii) Če je $\boldsymbol{c} \in \mathbb{F}_2^n[D]$ in $\boldsymbol{c} = \boldsymbol{x}G$, potem je $\boldsymbol{x} \in \mathbb{F}_2^k[D]$.* □

Trditev *(iii)* pravi, da če je $G(D)$ bazični kodirnik in je izhod polinomski, je takšen tudi vhod.
Za opis dodatnih ekvivalentnih trditev pa vpeljimo še pojem *stopnje vektorja* $\boldsymbol{v}(D)$ iz $\mathbb{F}^n[D]$. Le-ta je enaka največji stopnji komponent tega vektorja. (Opozorimo tudi, da je takšna definicija v skladu z definicijo "stopnje" matrike.) Zdaj lahko formuliramo tudi naslednjo karakterizacijo.

**Izrek 4.7.5.** *Naj bo $G(D)$ $k \times n$ polinomska matrika in naj bo $\boldsymbol{g}_i(D)$ njena $i$-ta vrstica. Naslednje trditve so ekvivalentne.*

*(i) Matrika $G(D)$ je reducirana.*

*(ii)* intdeg $G(D) =$ extdeg $G(D)$.

*(iii) Za poljuben $\boldsymbol{x}(D) = (x_1(D), \ldots, x_k(D)) \in \mathbb{F}_2^k[D]$, velja*

$$\deg(\boldsymbol{x}(D)G(D)) = \max_{1 \le i \le k} \{\deg\ x_i(D) + \deg\ \boldsymbol{g}_i(D)\}.$$

Lastnosti $(ii)$ pravimo *predvidljiva lastnost stopnje* reduciranih matrik.
Zelo prikladno bi bilo, če bi veljalo, da je vsaka polinomska generatorska matrika $(n, k)$ konvolucijske kode kanonična natako takrat, ko je bazična in reducirana. Na srečo je tako karakterizacijo možno dokazati. V pomoč nam bo priskočila naslednja lema.

**Lema 4.7.6.** *Naj bo $G(D)$ $k \times n$ polinomska matrika nad $\mathbb{F}_2(D)$ $k \le n$. Naj bo $N$ nesingularna $k \times k$ polinomska matrika z elementi iz $\mathbb{F}_2[D]$. Veljajo naslednje trditve:*

*(i)* intdeg $NG =$ intdeg $G +$ deg det $N$.

*(ii)* intdeg $G \le$ intdeg $NG$. *Enačaj velja natanko takrat, ko je $N$ unimodularna.*

*(iii)* intdeg $G \le$ extdeg $G$.

**Dokaz.**

(i) Opazimo najprej, da so $k \times k$ podmatrike matrike $NG$ natanko podmatrike matrike $G$ pomnožene z $N$ z leve. Torej so $k \times k$ minorji matrike $NG$ enaki $k \times k$ minorjem matrike $G$ pomnoženim z det $N$. S tem je dokaz za $(i)$ končan.

(ii) Sledi neposredno iz $(i)$.

(iii) Označimo z $m_i$ stopnjo $i$-te vrstice matrike $G$. Potem je zunanja stopnja enaka extdeg $G = m_1 + m_2 + \ldots + m_k$. Vsak $k \times k$ minor matrike $G$ je vsota ustreznih produktov elementov $G$, ki imajo v vsakem produktu po en faktor v vsaki vrstici. Od tod lahko sklepamo, da je stopnja produkta z najvišjo stopnjo največ $m_1 + m_2 + \ldots + m_k$, kar pomeni, da je to največja stopnja minorja. Od tod pa že dobimo intdeg $G \le$ extdeg $G$.

□

Po vseh zgornjih pripravah lahko dokažemo centralni izrek tega razdelka, ki se glasi

**Izrek 4.7.7.** *Polinomska generatorska matrika $(n, k)$ konvolucijske kode $C$ je kanonična če in samo če je bazična in reducirana.*

**Dokaz.**
($\Rightarrow$) Naj bo $G(D)$ kanonična generatorska matrika za $C$. Ker vemo, da imajo bazične generatorske matrike skupno notranjo stopnjo, le-to označimo z $d_0$. Nato med bazičnimi generatorskimi matrikami izberemo tako matriko $G_0(D)$, katere zunanja stopnja je čim manjša. Če z $U(D)$ označimo poljubno unimodularno matriko, lema 4.7.6 pove, da velja intdeg $UG_0 =$ intdeg $G_0 = d_0$. Po definiciji $G_0$ je extdeg $UG_0 \geq$ extdeg $G_0$ (ker $UG_0$ generira kodo $C$), kar pa že pomeni, da je $G_0$ reducirana. Ker pa ima $G_0$ najmanjšo možno notranjo stopnjo med polinomskimi generatorskimi matrikami konvolucijske kode $C$, je intdeg $G_0 \leq$ intdeg $G$. Iz leme 4.7.6 pa sklepamo, da je intdeg $G \leq$ extdeg $G$. Iz kanoničnosti matrike $G$ sledi extdeg $G \leq$ extdeg $G_0$. Torej

$$\text{intdeg } G_0 \leq \text{intdeg } G \leq \text{extdeg } G \leq \text{extdeg } G_0.$$

Iz prej dokazanega dejstva, da je $G_0$ reducirana in izreka 4.7.5 velja

$$\text{intdeg } G_0 = \text{extdeg } G_0.$$

Zato imamo v prejšnji neenačbi povsod enačaj. Zato je intdeg $G =$ intdeg $G_0 = d_0$, kar pomeni, da ima $G$ najmanjšo notranjo stopnjo med vsemi polinomskimi generatorskimi matrikami kode $C$ in je zato tudi bazična. Ker je tudi intdeg $G =$ extdeg $G$, je matrika $G$ tudi reducirana.
($\Leftarrow$) Recimo, da je $G$ bazična in reducirana polinomska generatorska matrika kode $C$. Naj bo $G_0$ poljubna polinomska generatorska matrika kode $C$. Lema 4.7.6 pove, da je extdeg $G_0 \geq$ intdeg $G_0$. Ker je $G$ bazična, je intdeg $G_0 \geq$ intdeg $G$. Vemo tudi, da je $G$ reducirana in zato intdeg $G =$ extdeg $G$ po izreku 4.7.5. S kombiniranjem omenjenih neenakosti dobimo extdeg $G_0 \geq$ extdeg $G$, od koder sledi, da je $G$ kanonična. □

Izrek 4.7.7 pokaže, da je minimalna notranja stopnja poljubne polinomske generatorske matrike $(n, k)$ konvolucijske kode $C$ tudi minimalna zunanja stopnja poljubne polinomske generatorske matrike kode $C$, torej je enaka vsoti vrstičnih stopenj kanonične generatorske matrike. Kar pa na prvi pogled ni očitno, je dejstvo, da je množica vrstičnih stopenj kanonične generatorske matrike enolično določena. Ta trditev bo trivialno sledila iz naslednje leme.

**Lema 4.7.8.** *Naj bo $G$ kanonična generatorska matrika $(n, k)$ konvolucijske kode $C$. Preuredimo vrstice matrike $G$ tako, da bodo stopnje vrstic zadoščale relaciji $m_1 \leq m_2 \leq \ldots \leq m_k$, kjer je $m_i$ stopnja $i$-te vrstice matrike $G$. Naj bo še $G'$ poljubna polinomska generatorska matrika za $C$. Analogno preuredimo vrstice te matrike tako, da stopnje vrstic $m'_i$ zadoščajo $m'_1 \leq m'_2 \leq \ldots \leq m'_k$. Potem velja $m_i \leq m'_i$ za $1 \leq i \leq k$.*

**Dokaz.** Naj bo $\boldsymbol{g}_i(D)$ $i$-ta vrstica matrike $G(D)$, za katero je $\deg \boldsymbol{g}_i(D) = m_i$. Analogno naj bo $\boldsymbol{g'}_i(D)$ $i$-ta vrstica matrike $G'$, za katero je $\deg \boldsymbol{g'}_i(D) = m'_i$. Predpostavimo, da trditev iz leme ne velja. Potem obstaja celo število $j$, za katerega veljajo neenakosti $m_1 \leq m'_1, \ldots, m_j \leq m'_j$ in $m_{j+1} > m'_{j+1}$. Naj bo $1 \leq i \leq j+1$. V tem primeru velja $\deg \boldsymbol{g'}_i(D) < m_{j+1}$. Vemo, da je $\boldsymbol{g'}_i(D)$ polinomska kodna beseda oz. "polinomski izhod", ter da je $G(D)$ bazična matrika. Torej mora obstajati tak "polinomski vhod" $\boldsymbol{x}_i(D) = (x_{i1}(D), \ldots, x_{ik}(D))$, da po izreku 4.7.4 velja $\boldsymbol{g'}_i(D) = \boldsymbol{x}_i(D)G$. $G$ je tudi reducirana, zato iz "predvidljive stopenjske lastnosti" iz izreka 4.7.5 sledi

$$\begin{aligned} m'_i = \deg \boldsymbol{g'}_i(D) = \deg \boldsymbol{x}_i(D)G(D) &= \max_{1\leq \ell \leq k} \{\deg x_{i\ell}(D) + \deg \boldsymbol{g}_\ell(D)\} \\ &= \max_{1\leq \ell \leq k} \{\deg x_{i\ell}(D) + m_\ell\}. \end{aligned}$$

Iz $\deg \boldsymbol{g'}_i(D) < m_{j+1}$ sledi $\deg x_{il}(D) = -\infty$ za $\ell \geq j+1$. Z drugimi besedami, velja $x_{i\ell}(D) = 0$ za $\ell \geq j+1$. Ker je tudi $\boldsymbol{g'}_i(D) = \boldsymbol{x}_i(D)G$ mora $\mathbf{g'}_i(D)$ biti polinomska kombinacija prvih $j$ vrstic matrike $G$ za $1 \leq i \leq j+1$. Prvih $j+1$ vrstic matrike $G$ je torej linearno odvisnih nad $\mathbb{F}_2(D)$, kar pa je protislovje. □

Pravkar dokazana lema nam dá naslednji rezultat.

**Izrek 4.7.9.** *Naj bo $C$ neka $(n, k)$ konvolucijska koda. Množica vrstičnih stopenj je enaka za vse kanonične generatorske matrike kode $C$.* □

Tej enolično določeni množici vrstičnih stopenj kanonične generatorske matrike kode $C$ pravimo množica *Forneyejevih indeksov kode C*. Vsota Forneyevih indeksov je enaka zunanji stopnji kanonične generatorske matrike, ki je tudi stopnja kode $C$, ki je v bistvu isto kot skupna dolžina kode $C$. Če z $m$ označimo stopnjo kode, lahko kodo označimo s trojico $(n, k, m)$. Viterbijev algoritem zahteva $2^m$ stanj, če uporabljamo kanoničen kodirnik. V primeru, ko za konstrukcijo kodirnika uporabimo kanonično generatorsko matriko, je spomin kodirnika enak največjemu Forneyevemu indeksu. Iz leme 4.7.8 vidimo, da je spomin kodirnika, izpeljanega iz poljubne polinomske generatorske matrike, večji ali enak spominu kode. S tem sklepom končujemo razdelek o kanoničnih matrikah in kodirnikih.

# Poglavje 5

# Odkodiranje

## 5.1 Viterbijevo odkodiranje s trdim odločanjem

V tem poglavju si oglejmo najbolj znan postopek za odkodiranje sporočil, ki so zakodirana s konvolucijsko kodo in prenešena po nekem komunikacijskem kanalu. Našel ga je Viterbi leta 1967.

Za začetek predpostavimo, da smo zakodirali sporočilo $\boldsymbol{u}_i = (u_i(1), \ldots, u_i(k))$ za $i = 0, 1, \ldots, L-1$ s pomočjo generatorske matrike $G$ in pri tem dobili kodno besedo $\boldsymbol{v}_i = (v_i(1), \ldots, v_i(n))$ za $i = 0, 1, \ldots, L+m-1$. Hkrati recimo, da smo prejeli sporočilo $\boldsymbol{r}_i = (r_i(1), \ldots, r_i(n))$ za $i = 0, 1, \ldots, L+m-1$.

Najprej definirajmo *težo poti* v mrežnem diagramu. Naj bo $e$ povezava med stanjema $s$ v času $i-1$ in $s'$ v času $i$. *Teža povezave* $e$ je Hammingova razdalja med oznako na povezavi $e$ in kosom $r(i-1)$ prejetega sporočila v času $i-1$. *Teža poti* $P$ je potemtakem vsota tež vseh povezav poti $P$. Torej sta teži povezave in poti odvisni od prejetega vektorja, ki predstavlja sporočilo, kot ga poznamo, potem ko smo ga prejeli preko komunikacijskega kanala.

Označimo zdaj z $a_0$ ničelno stanje v času 0. Naj bo $P$ pot, katere začetek je v $a_0$ in konec v $s$ v času $i$. Taki poti bomo rekli, da je *preživela v času* $i$, če je njena teža najmanjša med vsemi potmi z začetkom v $a_0$ in koncem v stanju $s$ v času $i$. Množico vseh v času $i$ preživelih poti, ki se začnejo v $a_0$ in končajo v stanju $s$ v času $i$ označimo s $S(s, i)$.

Vpeljimo še nekaj drugih oznak in pojmov. Naj bo $P$ pot z začetkom v $a_0$ in koncem v času $I$. Označimo s $y(P)$ *kodno besedo, pridruženo* $P$, kjer je $y_P(i)$ oznaka povezave v $P$, ki povezuje stanji v času $i$ in času $i+1$ za $0 \leq i < I$. Ustrezno naj bo $u(P)$ *sporočilo, pridruženo* $P$, kjer je $u_P(i)$ tisti vhodni kos informacije, ki na poti $P$ pripelje iz stanja v času $i$ v stanje v času $i+1$ za $0 \leq i < \min[I, L]$.

Zdaj imamo na voljo vse potrebno za predstavitev Viterbijevega algoritma.

**Viterbijev algoritem**

I. Nariši ustrezen mrežni diagram za $G$ in zamenjaj oznake na povezavah s težo povezav. Označi ničelno stanje (npr. z $a$).

II. Izračunaj $S(s,1)$ za vsa stanja na mrežnem diagramu iz prejšnjega koraka.

III. Za $i = 2, 3, \ldots, L+m$ iz $S(s, i-1)$ izračunaj $S(s,i)$ za vsa stanja $s$ na naslednji način: za vsako stanje $s'$ in vsako povezavo $e$ iz $s'$ v $s$, sestavi pot $P$ tako, da poti $P'$ dodaš povezavo $e$, kjer je $P' \in S(s', i-1)$. $P$ dodaj v $S(s,i)$, če je najmanjše teže med vsemi na omenjeni način sestavljenimi potmi.

IV. Najbližji sosed $v$ je katerikoli $y_P$ za $P \in S(s, L+m)$, dobljen iz sporočila $u(P)$.

**Trditev 5.1.1.** *Viterbijev algoritem pravilno poišče najbližjo kodno besedo.*

**Dokaz.** Najbližji sosed $y$ za $v$ je $y_P$, ki ga dobimo za neko pot $P \in S(a, L+m)$, saj vsebuje $S(a, L+m)$ vse najkrajše poti od $a_0$ do $a_{L+m}$ z $L+m$ povezavami ($a_{L+m}$ je ničelno stanje v času $L+m$. Torej je potrebno preveriti le to, da koraka II in III pravilno izračunata $S(a, L+m)$. To pa bo zagotovo držalo, če dokažemo, da koraka II in III pravilno izračunata $S(s,i)$. Korak II po definiciji rezultira v $S(s,1)$ za vsa stanja $s$. Če vzamemo $P \in S(s,i)$, je le-ta sestavljen iz poti $P'$, ki se konča v času $i-1$ v stanju $s'$, in povezave $e$ iz stanja $s'$ v stanje $s$. Recimo, da $P'$ ne pripada $S(s', i-1)$. Potem obstaja pot $P''$ od $a_0$ do $s'$ v času $i-1$ z manjšo težo. Potem pa ima tudi pot sestavljena iz $P''$ in povezave $e$, ki konča v stanju $s$ v času $i$, manjšo težo kot $P$. To pa je protislovje. Od tod sledi, da korak III pravilno določi $S(s,i)$, s tem pa je dokazana tudi pravilnost algoritma. □

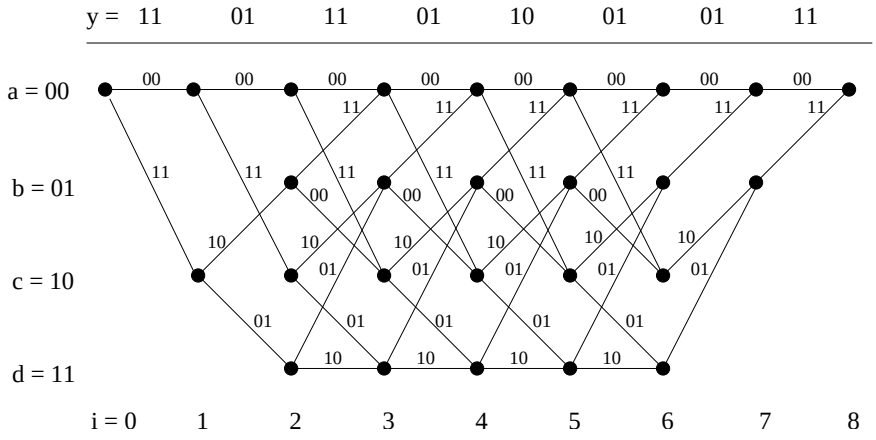
Iz pravkar povedanega sledi, da je v primeru trdega odločanja Viterbijev algoritem ekvivalenten iskanju prejetemu sporočilu najbolj verjetnega kandidata znotraj dane množice kodnih besed. Temu postopku pravimo tudi *iskanje najbližjega soseda*. Pripomnimo, da iz zgornje trditve ne sledi, da je odkodirano sporočilo res originalno sporočilo, ki je bilo poslano, saj je pravilnost odkodiranja odvisna tudi od števila napak, ki nastopijo med prenosom. Torej je odkodirano sporočilo enako kot originalno v primeru, ko pri prenosu ni prišlo do prevelikega števila napak.

**Primer:**

Izberimo konvolucijsko kodo z informacijsko stopnjo $R = 1/2$, spominom $m = 2$ in kodirno generatorsko matriko $G(D) = (1 + D + D^2 \;\; 1 + D^2)$. Predpostavimo hkrati, da komuniciramo preko binarnega simetričnega kanala, za katerega je $0 < \varepsilon < 1/2$ (torej odkodirnik uporablja princip trdega odločanja). Zaradi enostavnosti predpostavimo, da smo zakodirali le 6 vhodnih bitov in dodatni dve 0 na koncu (na ta način

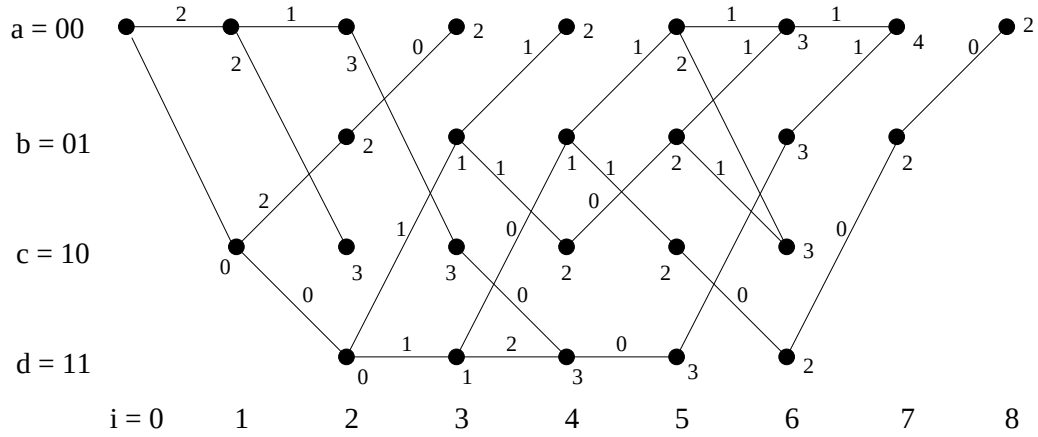
pripeljemo kodirnik v ničelno stanje; posledično, seveda, pa tudi kodo pretvorimo v bločno).

Predpostavimo, da smo prejeli sporočilo $r = 1101110110010111$. Uporabimo Viterbijev algoritem za odkodiranje. V ta namen bomo uporabili tudi mrežno predstavitev konvolucijske kode s kodirno matriko $G(D) = (1 + D + D^2 1 + D^2)$ (glej sliko 14). V vsakem časovnem trenutku na izhodu dobimo dva bita. Na sliki je pari prejetih bitov sporočila $r$ v vsakem trenutku ponazorjeni nad mrežo. Mreža za korak I je dana na naslednji sliki. Zaradi lažjega razumevanja bomo uporabili tudi naslednje oznake za stanja kodirnika: $a = 00, b = 01, c = 10$ in $d = 11$ ter $s_i$ za stanje $s$ v času $i$. Torej, ničelno stanje v času 0 je $a_0$.




Slika 14 - Mrežna predstavitev konvolucijske kode za matriko $G(D)$

Za korak II dobimo $S(a,1) = \{a_0a_1\}$, $S(b,1) = \emptyset$, $S(c,1) = \{a_0c_1\}$ in $S(d,1) = \emptyset$. Z uporabo koraka III poiščemo $S(s,2)$. Dobimo $S(a,2) = \{a_0a_1a_2\}$, $S(b,2) = \{a_0c_1b_2\}$, $S(c,2) = \{a_0a_1c_2\}$ in $S(d,2) = \{a_0c_1d_2\}$; takoj opazimo, da se vse poti v $S(s,2)$ začnejo z bodisi $\{a_0a_1\}$ bodisi $\{a_0c_1\}$, ki pa sta poti v $S(s',1)$. Poglejmo zdaj, kako dobimo $S(a,3)$. Pogledamo v vsak $S(s,2)$, od koder vidimo, da imamo le dve možnosti: prva je pot $a_0a_1a_2$, ki ji dodamo povezavo $a_2a_3$, kar nam dá $a_0a_1a_2a_3$, druga pa je pot $a_0c_1b_2$, ki ji dodamo povezavo $b_2a_3$, kar nam dá $a_0c_1b_2a_3$. Izračun teže poti nam pove, da ima prva pot težo 5, medtem ko ima druga pot težo 2. Zato je $S(a,3) = \{a_0c_1b_2a_3\}$. Podobno najprej izračunamo preostale $S(s,3)$, nato pa nadaljujemo do $S(a,8)$. Slika 15 kaže poti, ki se pojavijo med izračunom vseh $S(s,i)$; prav tako so v vsaki točki mreže dane tudi teže poti od $a_0$ do ustrezne točke. Dodajmo še, da obstajajo tudi poti, ki se končajo v neki vmesni točki (npr. $a_0a_1c_2$). To se zgodi, kadar je teža poti, sestavljena iz neke poti in še ene povezave, večja od teže neke druge poti, ki ima enako končno točko.




Slika 15 - Poti v mrežni predstavitvi pri odkodiranju

Uporabimo še korak IV Viterbijevega algoritma. Ugotovimo, da $S(a, 8)$ vsebuje samo pot $a_0c_1d_2d_3b_4c_5d_6b_7a_8$. Teža te poti je enaka 2. Kodna beseda $y_P$, ki jo preberemo sledeč poti $P$ v mreži na zadnji sliki, je 1101100100010111, ki se na dveh mestih (teža poti!) razlikuje od prejetega sporočila. Nato pogledamo ustrezne vhode za prvih 6 bitnih parov iz $y_P$ (zadnja dva para nas ne zanimata, ker vemo, da smo dodali dve ničli na koncu, zato da bi pripeljali kodirnik v ničelno stanje) in preberemo sporočilo 111011, ki je naše originalno sporočilo. ◊

Viterbijev postopek za odkodiranje v primeru mehkega odločanja pa se nekoliko razlikuje od zgoraj opisanega. Poglejmo si še ta postopek.

## 5.2 Viterbijevo odkodiranje z mehkim odločanjem

Viterbijev algoritem z uporabo mehkega odločanja je v svojem bistvu enak algoritmu z uporabo trdega odločanja. Ključna razlika se skriva v definiciji teže povezav. Ko te teže ustrezno postavimo, uporabimo enak algoritem z neznatnimi spremebami. Oglejmo si kako pridemo do omenjenih razlik in kaj je potrebno spremeniti, da dobimo postopek za odkodiranje z uporabo mehkega odločanja. Pri tem se bomo sklicevali na opis diskretnega kanala brez spomina s dvojišim vhodom in osmiškim izhodom.

Denimo, da vhodno sporočilo $\boldsymbol{u}_i = (\boldsymbol{u}_i^{(1)}, \ldots \boldsymbol{u}_i^{(k)}), i = 0, 1, \ldots, L-1$ zakodiramo z uporabo generatorske matrike $G$, ki opisuje $(n, k)$ konvolucijsko kodo. Na ta način dobimo kodno besedo $\boldsymbol{c}_i = (\boldsymbol{c}_i^{(1)}, \ldots \boldsymbol{c}_i^{(n)}), i = 0, 1, \ldots, L+m-1$. Označimo še, tako kot prej, stanje $s$ v času $i$ z $s_i$ ter z $a$ ničelno stanje kodirnika, ki je začetno in tudi končno stanje. Sporočilu dodamo tudi $m$ blokov po $k$ ničel na koncu, zato da kodirnik pripeljemo v ničelno stanje v času $L+m$. Bite kodne besede premešamo in dobljeni bitni tok potem moduliramo, prenesemo po kanalu in na drugem koncu prejmemo

v obliki signala $v(t)$, ki ga nato demoduliramo. Potem ko demodulirani prejeti bitni tok premešamo v nasprotni smeri, kot smo to naredili po kodiranju, dobimo prejeto sporočilo v vektorski obliki $\boldsymbol{v}_i = (\boldsymbol{v}_i^{(1)}, \ldots \boldsymbol{v}_i^{(n)}), i = 0, 1, \ldots, L + m - 1$, kjer je $\boldsymbol{v}_i \in \{0_1, 0_2, 0_3, 0_4, 1_1, 1_2, 1_3, 1_4\}$.

Ker bo Viterbijev algoritem z mehkim odločanjem prav tako uporabljal princip najbližjega soseda, je naloga algoritma poiskati tako kodno besedo $c$, ki maksimizira verjetnost $P(y \mid c)$. Ker pa je uporabljeni kanal brez spomina, lahko pišemo

$$P(\boldsymbol{v} \mid \boldsymbol{c}) = \prod_{i=0}^{L+m-1} \prod_{j=1}^{n} P(\boldsymbol{v}_i^{(j)} \mid \boldsymbol{c}_i^{(j)}).$$

Iskanje maksimuma tega izraza je ekvivalentno iskanju maksimuma njegovega logaritma

$$\ln P(\boldsymbol{v} \mid \boldsymbol{c}) = \sum_{i=0}^{L+m-1} \sum_{j=1}^{n} \ln(P(\boldsymbol{v}_i^{(j)} \mid \boldsymbol{c}_i^{(j)})),$$

katerega prednost je v tem, da nam zamenja produkt z vsoto (spomnimo se, da je Viterbijev algoritem v prejšnjem poglavju uporabljal vsoto tež povezav za izračun teže poti). Idejo in izpeljavo, ki sledi, je dal Massey. Z njo bomo dosegli pretvorbo tež povezav v cela nenegativna števila.

Maksimiziranje $\ln P(\boldsymbol{v} \mid \boldsymbol{c})$ je enako maksimiziranju izraza

$$\sum_{i=0}^{L+m-1} \sum_{j=1}^{n} \mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)}),$$

kjer je

$$\mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)}) = A(\ln(P(\boldsymbol{v}_i^{(j)} \mid \boldsymbol{c}_i^{(j)}) - f_{i,j}(\boldsymbol{v}_i)),$$

pri čemer je $A$ pozitivna konstanta in $f_{i,j}(\boldsymbol{v}_i)$ poljubna funkcija. Vrednost

$$\sum_{j=1}^{n} \mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})$$

je enaka teži povezave in ustreza Hammingovi razdalji, uporabljeni pri algoritmu s trdim odločanjem. Izračun si lahko olajšamo tako, da postavimo

$$f_{i,j}(\boldsymbol{v}_i) = \min_{c \in [0,1]} \{\ln(P(\boldsymbol{v}_i^{(j)} \mid \boldsymbol{c}))\}$$

in potem izberemo $A$ tako, da bo približek $\mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})$ pozitivno celo število. Na ta način imamo teže povezav določene. Z njihovo pomočjo lahko izračunamo tudi teže

poti. Od tu naprej pa nadaljujemo z Viterbijevim algoritmom iz prejšnjega poglavja, upoštevajoč, da so poti, ki preživijo, tiste z **največjo težo** in ne tiste z najmanjšo. To pa je tudi edina razlika med algoritmoma s trdim in mehkin odločanjem.

Uporabimo pravkar opisani Viterbijev algoritem z mehkim odločanjem za primer prenosa podatkov preko binarnega simetričnega kanala s prehodno verjetnostjo $\rho$. V tem primeru imamo

$$P(\boldsymbol{v} \mid \boldsymbol{c}) = \prod_{i=0}^{L+m-1} \prod_{j=1}^{n} P(\boldsymbol{y}_i^{(j)} \mid \boldsymbol{c}_i^{(j)})$$

$$= \prod_{i=0}^{L+m-1} \prod_{j=1}^{n} \rho^{d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})} (1-\rho)^{1-d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})}.$$

Od tod in iz zgornje izpeljave pa že dobimo

$$\mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)}) = A(\ln(\rho^{d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})} (1-\rho)^{1-d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})}) - f_{i,j}(\boldsymbol{v}_i))$$

$$= d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})(A \ln \frac{\rho}{1-\rho}) + A \ln(1-\rho) - A f_{i,j}(\boldsymbol{v}_i).$$

Če izberemo zdaj $A = -(\ln \frac{\rho}{1-\rho})^{-1}$, kar je pozitivno za $\rho < 1/2$, in $f_{i,j}(\boldsymbol{v}_i) = \ln(\rho)$, dobimo

$$\mu(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)}) = 1 - d_H(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)}).$$

Od tod sledi, da so teže povezav v mreži enake $n - d(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})$. Pot z maksimalno težo, merjeno na ta način, je torej natanko tista pot z minimalno težo, katere teže povezav merimo z $d(\boldsymbol{v}_i^{(j)}, \boldsymbol{c}_i^{(j)})$. To pa pomeni, da sta rezultata Viterbijevega algoritma z mehkim in trdim odločanjem enaka, če za prenos podatkov uporabimo binarni simetrični kanal, ki pa je tudi najpogosteje uporabljan kanal.

## 5.3 Ocene bitnih napak pri odkodiranju

V praksi Viterbijev algoritem odkodira dolga zaporedja bitov še preden odkodirnik pripeljemo nazaj v ničelno stanje z $mk$ ničelnimi biti, ki jih spravimo v kodirnik za zaključevanje informacijskega zaporedja. Napačno odkodirana pot v mrežni predstavitvi se v splošnem združi s pravilno potjo še preden dosežemo konec. Napaka je ponavadi predstavljena kot "grozd" napačno odkodiranih informacijskih simbolov, ki se nekje začne in nekje konča. Potrebno je vedeti, da se grozd vedno začne in konča z napako. Prav tako velja, da v primerih, ko imamo $mk$ ali več zaporednih pravilno odkodiranih informacijskih simbolov, se je napačna pot združila s pravo

potjo, preden jo je znova zapustila. Pravimo, da se je zgodila *večkratna napaka*, sestavljena iz ločenih grozdov napak.

Pri bločnih kodah se kot merilo za verjetnost nastopa napak uporablja *verjetnost bločne napake*, ki pove, kakšna je verjetnost za nastop napake za vsak blok posebej. Za ocenjevanje napak pri konvolucijskih kodah z uporabo Viterbijevega algoritma takšna ocena ni primerna, saj je zaradi uporabe zelo dolgih zaporedij bitov tovrstna verjetnost zelo blizu vrednosti 1, ne glede na ustrezno sistemsko zaščito informacijskih simbolov. Zato se pri odkodiranju z Viterbijevim algoritmom uporablja *verjetnost bitne napake* $P_b$ kot mera za oceno zanesljivosti digitalnega komunikacijskega sistema, ki ga uporabljamo za prenos informacij, zakodiranih s konvolucijskimi kodami. Verjetnost bitne napake je tako lastnost kode kot tudi lastnost kodirne matrike. Odvisna je od preslikave, ki informacijskemu zaporedju priredi kodno besedo. Dodatni pojem, ki ga bomo uporabili, je *verjetnost grozdne napake* $P_B$, ki nam bo povedala, kakšna je verjetnost, da se grozdna napaka začne v danem vozlu mreže. Verjetnost grozdne napake je lastnost kode in ji ponekod pravijo tudi *verjetnost prvega nastopa napake* ali *verjetnost napake vozla*. Naše izpeljave bomo začeli z ocenjevanjem verjetnosti grozdne napake, saj je te ocene nekoliko lažje izpeljati kot ocene za verjetnosti bitnih napak. Prav tako bomo izpeljali samo zgornje ocene, saj so te z uporabniškega stališča nekoliko bolj pomembne kot spodnje.

Dejstvo je, da z dodajanjem vedno novih poti v brajdnik ne moremo doseči slabše ocene. Zato je verjetnost grozdne napake za konvolucijsko kodo, sestavljeno iz končnih kodnih besed, navzgor omejena z verjetnostjo grozdne napake za neskončno dolge kodne besede.

### 5.3.1 Verjetnost grozdne napake

Verjetnost grozdne napake *ni* enaka za vse vozle na pravilni poti. Oglejmo si, zakaj to drži. Predpostavimo, da se grozd začne v času $i, i > 0$. Ponavadi se ta grozd ni začel z nekim dogodkom, ki bi vseboval veliko kanalnih napak na začetku, saj bi tak dogodek povzročil, da bi bil začetek grozda na neki manjši globini. Od tod sledi, da verjetnost grozdne napake na globini $i, i > 0$ ne more bit večja od tiste v korenu mreže. Naša ocena bo veljala za vsa vozlišča.

Imejmo zdaj konvolucijsko kodo, ki jo uporabljamo za komunikacijo prek binarnega simetričnega kanala z verjetnostjo preskoka $\varepsilon, 0 < \varepsilon < 1/2$. Predpostavimo, da smo preko kanala poslali zakodirano ničelno sporočilo (sporočilo, sestavljeno iz samih ničel). Ko besedilo sprejmemo, je v njem nekaj neničelnih bitov (torej je prišlo do napake pri prenosu). Oceniti hočemo verjetnost, da odkodirnik vrne neko neničelno zaporedje kot približek za poslano sporočilo. Recimo, da je razdalja med (napačno) odkodirano in pravilno kodno besedo enaka $d$. Ločiti moramo med primeroma, ko je $d$ lih ali sod, saj pri sodem $d$ ni nujno, da pri odkodiranju naredimo napako v

primeru, ko imamo $d/2$ napak med $d$ biti. Torej bomo v takih primerih pravilno pot odvrgli z verjetnostjo $1/2$. Zato je verjetnost $p_d$, da imamo $d/2$ napak v $d$ zaporednih bitih pri lihem $d$ enaka

$$p_d = \sum_{e=(d+1)/2}^{d} \binom{d}{e} \varepsilon^e (1-\varepsilon)^{d-e}$$

pri sodem pa

$$p_d = \frac{1}{2}\binom{d}{d/2}\varepsilon^{d/2}(1-\varepsilon)^{d/2} + \sum_{e=(d+1)/2}^{d} \binom{d}{e} \varepsilon^e (1-\varepsilon)^{d-e}.$$

Izraz $\varepsilon^e(1-\varepsilon)^{d-e}$ narašča, ko $e$ pada. Za lih $d$ potem velja:

$$p_d = \sum_{e=(d+1)/2}^{d} \binom{d}{e} \varepsilon^e (1-\varepsilon)^{d-e} < \sum_{e=(d+1)/2}^{d} \binom{d}{e} \varepsilon^{d/2} (1-\varepsilon)^{d/2}$$

$$= \varepsilon^{d/2}(1-\varepsilon)^{d/2} \sum_{e=(d+1)/2}^{d} \binom{d}{e} < \varepsilon^{d/2}(1-\varepsilon)^{d/2} \sum_{0}^{d} \binom{d}{e}$$

$$= \left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^d$$

Izkaže se, da je $\left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^d$ zgornja meja tudi v primeru, ko je $d$ sod. Tej meji pravimo *Bhattacharyyajeva meja* in pišemo

$$p_d < \left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^d = z^d.$$

Ta ocena velja za vse $d$. Parametru $z$ pravimo *Bhattacharyyajev parameter*.

Označimo zdaj z $E^{(k)}$ dogodek, da se grozdna napaka začne v korenu mreže in izhaja iz poti $k$. Velja

$$P_B \leq P(\bigcup E^{(k)}) \leq \sum P(E^{(k)}).$$

Pri tem vzamemo unijo in vsoto po vseh nepravilnih poteh, ki izhajajo iz korena.

Konvolucijske kode so linearne. Zato lahko brez izgube splošnosti predpostavimo, da je pravilna pot ničelna. Potem pri predpostavki, da je Hammingova teža $k$-te napačne poti enaka $d$, imamo

$$P(E^{(K)}) = p_d$$

Od tod pa že sledi, da je

$$P_B \leq \sum_{d=d_{free}}^{\infty} n_d p_d,$$

kjer je $n_d$ število poti s težo $d$, torej težni spekter kode, $d_{free}$ pa prosta razdalja kode. Število poti s težo $d$ za $d = d_{free}, d_{free+1}, \ldots$ je dano s pomočjo števca tež poti in je enako

$$T(W) = \sum_{d=d_{free}}^{\infty} n_d W^d$$

S kombinacijo vsega dognanega lahko formuliramo sledeči izrek.

**Izrek 5.3.1.** *Verjetnost grozdne napake ob uporabi konvolucijske kode za komunikacijo prek binarnega simetričnega kanala z verjetnostjo preskoka $\varepsilon$ in odkodiranjem po principu najverjetnejšega kandidata je navzgor omejena z*

$$P_B < \sum_{d=d_{free}}^{\infty} n_d (2\sqrt{\varepsilon(1-\varepsilon)})^d = T(W)|_{W=2\sqrt{\varepsilon(1-\varepsilon)}},$$

*kjer je $T(W)$ števec tež poti generatorske matrike $G(D)$.* □

Ta rezultat je Van de Meeberg ([16]) dodatno poenostavil, potem ko je opazil, da velja enakost

$$p_{2i-1} = p_{2i}, i \geq 1,$$

torej, da je za vsako pot verjetnost napake pri odkodiranju za lih $d$ enaka kot, če $d$ povečamo za 1. Iz tega dobimo oceno

$$P_B < \Big(\frac{1+W}{2} T(W) + \frac{1-W}{2} T(-W)\Big)|_{W=2\sqrt{\varepsilon(1-\varepsilon)}}.$$

### 5.3.2 Verjetnost bitne napake

Končno se lahko posvetimo izpeljavi ocene za verjetnost bitne napake. Pri izpeljavi nam bo v veliko pomoč razširjeni števec poti.

Predpostavimo, da smo prejeli zaporedje $\boldsymbol{r} = \boldsymbol{r}_0 \boldsymbol{r}_1 \ldots \boldsymbol{r}_{K-1}$ $K$-tih $n$-teric, ga odkodirali v $\hat{\boldsymbol{v}} = \hat{\boldsymbol{v}}_1 \hat{\boldsymbol{v}}_2 \ldots \hat{\boldsymbol{v}}_{K-1}$ in dobili ustrezen približek informacijskega zaporedja $\hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}_1 \hat{\boldsymbol{u}}_2 \ldots \hat{\boldsymbol{u}}_{K-1}$, ki pa vsebuje $I(K)$ napak.

Naj $I$ označuje število napačno predvidenih simbolov v grozdu $k$-teric dolžine $L$ in naj $N$ označuje število $k$-teric med dvema grozdoma. Torej grozd $j, j = 0, 1, 2, \ldots$, vsebuje $I_j$ napačno predvidenih informacijskih simbolov, je sestavljen iz $L_j$ $k$-teric in ločen od prejšnjega grozda z $N_j$ $k$-tericami, kjer je $N_j > m, j = 0, 1, 2, \ldots$.

Napačno predvideni informacijski simboli, ki so ločeni z $m$ ali manj $k$-tericami po definiciji pripadajo istemu grozdu.

**Definicija 5.3.2. Verjetnost bitne napake** *$P_b$ definiramo kot razmerje med številom napačno predvidenih informacijskih simbolov in skupnim številom informacijskih simbolov.*

Po zakonu o velikih številih velja

$$\lim_{K\to\infty} P\Big(\Big|P_b - \frac{I(K)}{Kk}\Big| > \varepsilon\Big) = 0$$

za poljuben $\varepsilon > 0$, ali ekvivalentno

$$P_b = \lim_{K\to\infty} \frac{I(K)}{Kk} \quad \text{z verjetnostjo 1}$$

Recimo, da je $I(K)$ napak porazdeljenih med $J$ grozdi z $I_0, I_1, \ldots, I_{J-1}$ napakami, dolžin $L_0, L_1, \ldots, L_{J-1}$, in so ločeni z intervali dolžin $N_0, N_1, \ldots, N_{J-1}$, ki so brez napak. Iz zadnje limite potem sledi

$$P_b = \lim_{J\to\infty} \frac{\sum_{j=0}^{J-1} I_j}{k\sum_{j=0}^{J-1}(N_j + L_j)}$$

$$\leq \lim_{J\to\infty} \frac{\sum_{j=0}^{J-1} I_j}{k\sum_{j=0}^{J-1} N_j} \quad \text{z verjetnostjo 1}$$

in zato tudi

$$P_b \leq \frac{\lim_{J\to\infty}\sum_{j=0}^{J-1} I_j}{\lim_{J\to\infty} k\sum_{j=0}^{J-1} N_j} \quad \text{z verjetnostjo 1.}$$

Po zakonu o velikih številih je limita v imenovalcu enaka pričakovani vrednosti (matematičnemu upanju) števila bitnih napak v grozdu

$$E[I \mid \text{grozdna napaka}] = \lim_{J\to\infty} \frac{1}{J}\sum_{j=0}^{J-1} I_j \quad \text{z verjetnostjo 1,}$$

limita števca pa je enaka pričakovani vrednosti dolžine (merjene v številu povezav) intervalov brez napak

$$E[N] = \lim_{J\to\infty} \frac{1}{J}\sum_{j=0}^{J-1} N_j \quad \text{z verjetnostjo 1.}$$

Torej imamo

$$P_b \leq \frac{E[I \mid \text{grozdna napaka}]}{kE[N]}.$$

Verjetnost $P_B$, da se grozdna napaka začne v danem vozlišču lahko definiramo kot limito, ko $J \to \infty$ število grozdov $J$, deljeno s številom mest, kjer bi se grozd lahko začel. To število mest je manjše od $\sum_{j=0}^{J-1} N_j$. Zato imamo

$$P_B \geq \lim_{J\to\infty} \frac{J}{\sum_{j=0}^{J-1} N_j}$$

$$= \frac{1}{\lim_{J\to\infty} \sum_{j=0}^{J-1} N_j} = \frac{1}{E[N]} \quad \text{z verjetnostjo 1}$$

Iz vsega skupaj izpeljemo oceno

$$P_b \leq \frac{1}{k}\, E[I \mid \text{grozdna napaka}] P_B.$$

Označimo zdaj s $p(i)$ verjetnost, da grozd povzroči $i$ napak v predvidenem informacijskem zaporedju. Imamo

$$P_B = \sum_{i=1}^{\infty} p(i)$$

oziroma

$$\sum_{i=1}^{\infty} p(i \mid \text{grozdna napaka}) = 1,$$

kjer je

$$p(i \mid \text{grozdna napaka}) = \frac{p(i)}{P_B}$$

in

$$E[I \mid \text{grozdna napaka}] = \sum_{i=1}^{\infty} i\, p(i \mid \text{grozdna napaka}) = \frac{\sum_{i=1}^{\infty} ip(i)}{P_B}.$$

Končno dobimo

$$P_b \leq \frac{1}{k} \sum_{i=1}^{\infty} i\, p(i).$$

Naj bo zdaj $n(w, \ell, i)$ število poti s težo $w$ dolžine $\ell$, ki v približku informacijskega zaporedja vsebujejo $i$ napak. Iz Bhattacharayyave meje sledi, da je verjetnost, da pot teže $w$ povzroči napako pri odkodiranju, navzgor omejena z

$$\left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^w,$$

kjer je $\varepsilon$ verjetnost preskoka pri binarnem simetričnem kanalu. Potem z uporabo pravila za verjetnost unije dobimo

$$p(i) < \sum_{w=d_{free}}^{\infty} \sum_{\ell=\nu_{min}+1}^{\infty} n(w,\ell,i)\left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^w$$

kjer je $\nu_{min} = \min_i\{\nu_i\}$. Števila $n(w,\ell,i)$ so koeficienti razširjenega števca poti $T(W,L,I)$:

$$T(W,L,I) = \sum_w \sum_\ell \sum_i n(w,\ell,i) W^w L^\ell I^i.$$

S kombiniranjem zadnjih formul dobimo naslednji izrek.

**Izrek 5.3.3.** *Verjetnost bitne napake pri uporabi konvolucijske kode z generatorsko matriko $G(D)$ in spominom $m$ za komunikacijo preko binarnega simetričnega kanala z verjetnostjo preskoka $\varepsilon$ in odkodiranjem po principu najverjetnejšega kandidata je navzgor omejena z*

$$\begin{aligned} P_b &< \frac{1}{k} \sum_{w=d_{free}}^{\infty} \sum_{\ell=\nu_{min}+1}^{\infty} \sum_{i=1}^{\infty} i n(w,\ell,i)\left(2\sqrt{\varepsilon(1-\varepsilon)}\right)^w \\ &= \frac{1}{k} \frac{\partial T(W,L,I)}{\partial I}\Big|_{W=2\sqrt{\varepsilon(1-\varepsilon)},\, L=1,\, I=1} \end{aligned}$$

*kjer je $T(W,L,I)$ razširjeni števec poti generatorske matrike $G(D)$.* □

Van de Meeberg je tudi ta rezultat nekoliko izboljšal z uporabo izboljšave ocene zgornje meje verjetnosti grozdne napake:

**Izrek 5.3.4.** *Verjetnost bitne napake pri uporabi konvolucijske kode z generatorsko matriko $G(D)$ spomina $m$ za komunikacijo preko binarnega simetričnega kanala z verjetnostjo preskoka $\varepsilon$ in odkodiranjem po principu najverjetnejšega kandidata je navzgor omejena z*

$$P_b < \frac{1}{k}\frac{1+W}{2}\frac{\partial T(W,L,I)}{\partial I} + \frac{1-W}{2}\frac{\partial T(W,L,I)}{\partial I}\Big|_{W=2\sqrt{\varepsilon(1-\varepsilon)},\, L=1,\, I=1}$$

*kjer je $T(W,L,I)$ razširjeni števec poti generatorske matrike $G(D)$.* □

S temi ocenami zaključujemo poglavje o odkodiranju konvolucijskih kod. Omenimo še, da se omenjene zgornje meje dá še nekoliko izboljšati ter da je iz teh splošnih

izrekov možno izpeljati tudi ocene z enostavnejšimi formulami v primerih, ko poznamo natančne parametre komunikacijskega sistema. Prav tako se verjetnosti grozdne in bitne napake dá oceniti navzdol, a nas te ocene tukaj niso zanimale, saj je zgornja meja na nek način najpomembnejši podatek, ker nam dá občutek za zanesljivost postopka odkodiranja.

# Poglavje 6

# Konvolucijske kode v praksi

## 6.1 Uvod

Teorija kodiranja je s Shannonovim odkritjem iz leta 1948 postala ena izmed najbolj vročih teorij svojega časa. Teoretičnemu raziskovanju je sledilo zelo aktivno praktično uvajanje najnovejših dognanj. Posebej je uporaba teorije kodiranja v svojih zgodnjih dneh našla svoje mesto v vesoljskih komunikacijskih sistemih, kjer pasovna širina za prenos podatkov ni bila problematična. Prvi poskusi vpeljave rezultatov teorije kodiranja so bili usmerjeni v zmanjševanje zahtev po energiji, potrebni za prenos in stremljenje k dosegu Shannonove meje za kapaciteto kanala. Z razvojem telefonije in brezžičnih komunikacijskih sistemov je teorija kodiranja postala zelo zanimiva tudi tam, kjer je pasovna širina za prenos podatkov omejena, posebno v satelitski komunikaciji, mobilnem komuniciranju in prenosu govora. Zato bomo v tem poglavju opisali nekaj uporab konvolucijskih kod za preprečevanje napak v komunikaciji z vesoljskimi odpravami, satelitski komunikaciji, mobilni telefoniji in prenosu govora.

## 6.2 Raziskovanje vesolja

Pri prenosu podatkov iz vesolja se pogosto dogaja, da je prejeti signal izredno šibek. Napake so večinoma naključne in nastajajo zaradi belega Gaussovega šuma. Količina informacije v signalu je pogosto zelo velika in zaradi tega mora biti tudi možnost popravljanja napak čim večja. Pri tem nam gre na roko dejstvo, da pasovna širina ni omejena in nam to omogoča izgradnjo kompleksnih odkodirnikov. V uporabi so najpogostejše konvolucijske kode z nizko informacijsko stopnjo, ki jih odkodiramo s pomočjo zaporednih odkodirnikov. Za izboljšavo učinkovitosti pa so pogoste tudi metode sestavljanja kod (concatenation codes).

**Pioneer**

Vesoljska odprava *Pioneer 9* je odprava iz leta 1968, ki je svoje raziskave opravljala v Sončevi orbiti in je prva vesoljska odprava, kjer je bila za preprečevanje napak uporabljena konvolucijska koda. To je bila sistematična koda z informacijsko stopnjo 1/2 z 20-mestnim spominom. Minimalna Hammingova razdalja kode je bila 10. Za naslednje odprave pa so bile uporabljenje še zmogljivejše kode. Tako je bila na primer v odpravi *Pioneer 10* leta 1972, ki je raziskovala Jupiter, uporabljena koda enake informacijske stopnje, vendar pa s spominom velikosti 31 z minimalno Hammingovo razdaljo 11. Prav tako je za odkodiranje uporabljen novejši Fanov algoritem.

#### Voyager

Leta 1977 je na svojo raziskovalno odpravo oddaljenih planetov odšla sonda *Voyager*. Le-ta je dosegla Saturn leta 1979 in planet Jupiter leta 1981. Slike, ki jih je sonda pošiljala z obeh planetov, so bile zakodirane z dvema konvolucijskima kodama tipa (2, 1, 6) in (3, 1, 6). Na naslednji odpravi Voyagerja na Uran, leta 1986, srečamo tudi eno izmed prvih, danes zelo priljubljenih metod kodiranja: kombiniranje nebinarnih linearnih bločnih kod s konvolucijskimi, kar izboljša učinkovitost samega postopka kodiranja. Koda (2, 1, 6) je namreč uporabljena skupaj z Reed-Solomonovo kodo tipa (255, 223) nad končnim obsegom $GF(2^8)$ (za boljši vpogled v Reed-Solomonove kode glej [8]). Reed-Solomonova koda je uporabljeni kot zunanja, konvolucijska koda pa kot notranja koda. Za odkodiranje tako sestavljene kode sta uporabljena Viterbijev algoritem z mehkim odločanjem za konvolucijsko kodo in Massey-Berlekampov algoritem s trdim odločanjem (Massey-Berlekampov algoritem je podrobno opisan v [7] in skrčeno v [10]).

#### Galileo

Odprava Galileo iz leta 1989 je šla pri uporabi konvolucijskih kod še korak naprej. Tudi pri tej odpravi je bila uporabljena kombinacija (4, 1, 14) konvolucijske kode skupaj z (255, 223) Reed-Solomonovo kodo. Biti, kodirani na prvem in tretjem mestu so bili zamenjani, kar je pripomoglo predvsem pri zanesljivosti prenosa in sinhronizaciji simbolov. Vse skupaj je prineslo dodatna $2dB$ v primerjavi s kombinacijo, uporabljeno pri odpravi Voyager.

## 6.3 Satelitska komunikacija

Pri satelitski komunikaciji imamo anteno in moč omejene velikosti. Šum je ponovno beli Gaussov in napake so po svoji naravi naključne. Zaradi vsega omenjenega ponovno potrebujemo izredno močan sistem za preprečevanje in odpravljanje napak. Pri postavljanju tega sistema moramo upoštevati dejstvo, da nam je na voljo omejena

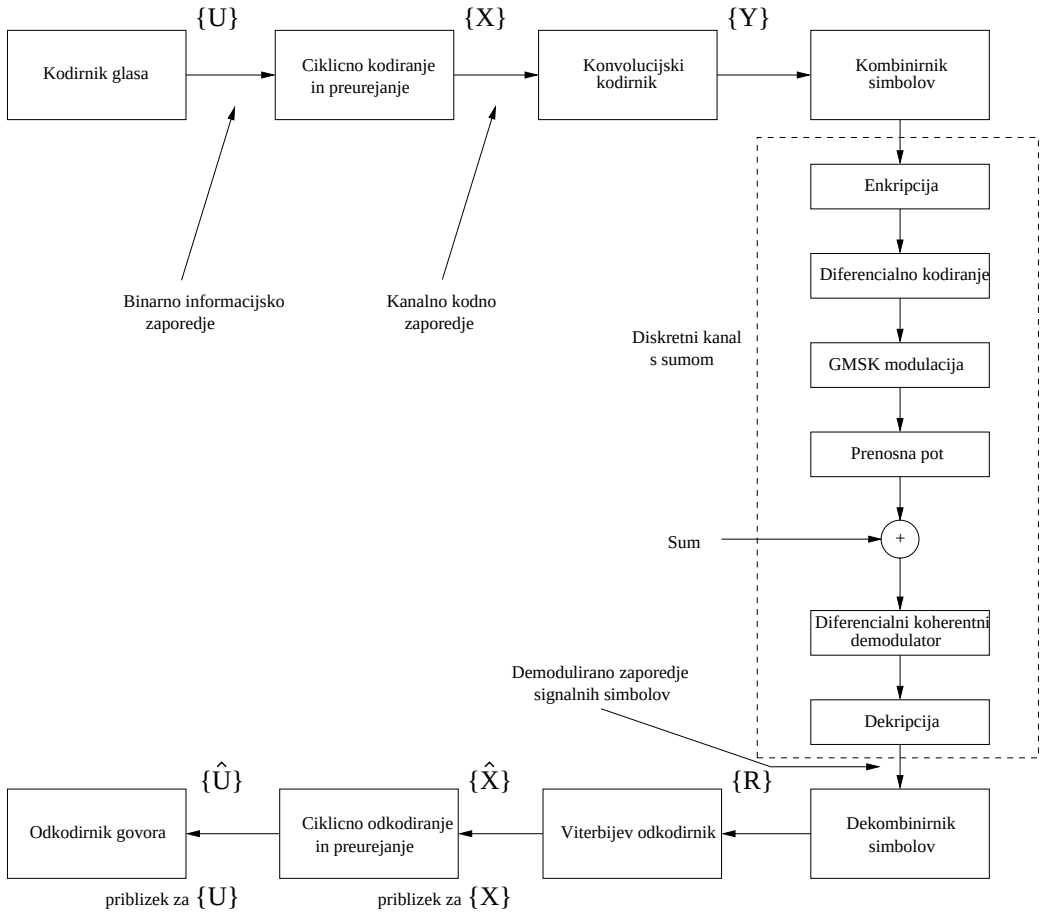
pasovna širina, kar nas prisili v implementacijo kode z visoko informacijsko stopnjo. Tako so leta 1969 za prenosni sistem digitalne barvne televizijke slike (DITEC) prek satelita INTELSAT IV uporabili (8, 7, 146) sistematično sebi-ortogonalno konvolucijsko kodo. Leta 1975 so kodo spremenili v kodo tipa (8, 7, 47), katere minimalna Hammingova razdalja je 5 in lahko popravi dve napaki.

## 6.4 Mobilna komunikacija

Osnovna značilnost sistemov za mobilno komuniciranje je velika omejenost pasovne širine in signalne prenosne moči. Prav tako sta omejeni velikosti mobilnega prenosnika in sprejemnika, zaradi česar je možno tudi izginjanje signala pri prenosu. V zadnjih 25 letih smo priča povečanemu raziskovanju in uporabi kod za odpravljanje napak v kombinaciji s celularno mobilno elektroniko. Največji napredek se je zgodil pri uvajanju celularnega mobilnega radia, ki maksimalno izrablja uporabno pasovno širino s ponovno uporabo frekvenc, ki so na voljo. Tak razvoj omogoča sledenje naraščajočim zahtevam po učinkovitem digitalnem mobilnem prenosu pri omejenih pasovnih širinah. Nekaj težav je zaradi medkanalne interference ter zaradi interference in izginjanja signala na sosednjih kanalih. V takih sistemih so zelo zaželjene konvolucijske kode, ki dobro preprečujejo napake in imajo manj kompleksne sisteme za odkodiranje.

### Sistem za digitalni GSM radio

Povečanje zanimanja za teorijo kodiranja je povzročila ustanovitev GSM komiteja (Groupe Speciale Mobile) pri Komiteju evropskih pošt in telekomov (CEPT) leta 1982. Prednostna naloga novoustanovljenega komiteja je bila poiskati bolj učinkovite sisteme za prenos signalov, kot so bili tedanji analogni sistemi. Ta zahteva je naravno izhajala iz povečanega zanimanja za mobilno telefonijo, ki je povzročila nezmožnost analognih sistemov za upravljanje z vedno večjim številom uporabnikov. Leta tako teoretskega kot tudi praktičnega raziskovanja so obrodila sadove s postavitvijo standarda in implementacijskih specifikacij panevropske celularne digitalne sheme iz leta 1988, znane kot "full-rate" GSM sistem. Na spodnji sliki lahko vidimo enega od takšnih tipičnih sistemov




Slika 16 - Full-rate GSM sistem

GSM digitalni radio sistem uporablja digitaliziran glas skupaj s tehnikami procesiranja digitalnega signala, ki izboljšujejo spektralno učinkovitost in integrabilnost. Za preprečitev grozdnih napak so v sistemu uporabljene kode za preprečevanje napak s kombiniranjem simbolov. Opis priporočil GSM komiteja in GSM digitalnega radia sta podrobno opisana v [12] in [13].

## 6.5 Prenos govora pri klasični telefoniji

Glasovni prenos pri klasični telefoniji poteka po kanalu, katerega širina je med $30Hz$ in $3400Hz$ in predstavlja tipičen primer sistema s kanalom omejene pasovne širine. To dejstvo vpliva na implementacijo večnivojske modulacije, namenjene boljšemu izkoristku spektralne učinkovitosti. S popularizacijo interneta je prišlo do izjemnega razvoja modemov, ki uporabljajo sisteme, zasnovane na prenosu govora. Ključni napredek pri tem razvoju predstavljajo metode, ki kombinirajo kodiranje z večnivojsko modulacijo. Leta 1983 je Mednarodni svetovalni telefonski in telegrafski komite (CCITT) priporočil uporabo 8-stanjskega rotacijsko invariantnega nelinearnega kon-

volucijskega kodiranja z odkodiranjem na principu mehkega odločanja pri iskanju najverjetnejšega kandidata. Deset let kasneje pa so to priporočilo razširili na konvolucijske kode s 16-, 32- in 64-imi stanji. Zelo podroben pregled razvoja modemov je podal Forney v [11].

# Literatura


[1] L. H. C. Lee, *Convolutional coding, fundamentals and applications*, Artech House Publishers. 1997

[2] R. Johannesson, K. Sh. Zigangirov, *Fundamentals of Convolutional Coding*, IEEE Press. 1998

[3] S. Klavžar, *O teoriji kodiranja, linearnih kodah in slikah z Marsa*, Obzornik za matematiko in fiziko 45 97-106 (1998).

[4] W. Cary Huffman, V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press. 2003

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

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

[7] T. Novak, *magistrsko delo, Berlekamp-Masseyev algoritem*, FMF 2003.

[8] S. Lin, D. J. Costello Jr, *Error Control Coding: Fundamentals and Applications*, Prentice-Hall. 1983

[9] C. E. Shannon, *A Mathematical Theory of Communication*, Bell Syst. Tech J. Vol. 27 No.3 str. 379-423 in Vol. 27 No.4 str. 623-656 (1948).

[10] A. Jurišić, A. Žitnik, *Reed-Solomonove kode*, Obzornik za matematiko in fiziko (200?).

[11] G. D. Forney Jr., *Coded Modulation for Band-Limited Channels*, IEEE Information Theory Society Newsletter, Vol. 40 str 1-7 (1990).

[12] GSM Recommendations 05.03, *Channel Coding*, Draft version 3.4.0 (1988).

[13] M. R. L. Hodges, *The GSM Radio Interface*, British Telecom Technology Journal, Vol. 8 No.1 str. 31-43 (1990).

[14] A. J. Viterbi, *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE Trans. on Information Theory, Vol. IT-13 No.2 str. 260-269 (1967).

[15] A. J. Viterbi, *Convolutional codes and their performance in communication systems*, IEEE Trans. on Communication Technology, COM-19 str. 751-772 (1971).

[16] L. Van de Meeberg, *A tightened upper bound on the error probability of binary convolutional codes with Viterbi decoding*, IEEE Trans. on Information Theory, IT-20 str. 389-391 (1974).

# Stvarno kazalo