Teorija kodiranja i linearni kodovi

Lukanović, Marina

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni nastavnički studij matematike i informatike

Marina Lukanović

Teorija kodiranja i linearni kodovi

Diplomski rad

Osijek, 2017.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Sveučilišni nastavnički studij matematike i informatike

Marina Lukanović

Teorija kodiranja i linearni kodovi

Diplomski rad

Mentor: izv. prof. dr. sc. Ivan Matić

Osijek, 2017.

Sadržaj

Uvod

U zadnje vrijeme teorija kodiranja prerasta u disciplinu kojoj se pridaje sve više pozornosti i koja se sve više razvija prvenstveno zbog toga što se može primjeniti u gotovo svakom području komunikacije od satelitskog prijenosa informacija do pohrane podataka u računalima. Cilj ovoga rada je dati kratak uvod u teoriju kodiranja te upoznati čitatelja sa osnovnim pojmovima vezanim uz ovu disciplinu.

U prvom poglavlju biti će opravdana potreba za razvojem teorije kodiranja. Vidjeti će se zašto je važno kodirati poruke tijekom slanja te će biti opisana dva najjednostavnija načina kodiranja sa svrhom nadziranja pogrešaka prilikom prenošenja informacija. Također, biti će dan opis Shannonovog općeg komunikacijskog sustava.

U drugom poglavlju upoznaje se čitatelja sa blok kodovima te osnovnim pojmovima vezanim uz njih te se daje primjer kako se na jednostavan način može dekodirati primljena poruka. Definira se i najmanja udaljenost koda koja je važna u određivanju broja grešaka koje se u kodu mogu prepoznati i ispraviti. Nakon toga govori se o Shannonovom teoremu koji opravdava postojanje dobrih kodova te se daje njegov iskaz.

U trećem poglavlju govori se o linearnim kodovima. Prije definiranja linearnih kodova navode se definicije polja i vektorskog prostora. Objašnjena su i dva načina na koja možemo opisati linearni kod: pomoću generirajuće matrice i matrice provjere parnosti zajedno sa primjerima te su dani osnovni teoremi vezani za ove kodove.Opisana je i metoda kodiranja pomoću generirajuće matrice koda te dekodiranja pomoću matrice provjere parnosti. Definira se i sindrom vektora koji predstavlja bitan faktor u dekodiranju primljene poruke i navodi se primjer dekodiranja koji potrkrepljuje tu tvrdnju. Dan je i teorem koji nam može poslužiti za određivanje minimalne udaljenosti linearnog koda ako je dana matrica provjere parnosti kao i za kreiranje jedan-ispravljajućih kodova. Nakon toga opisani su binarni Hammingovi kodovi koji imaju dosta zanimljiva svojstva te se često koriste u memoriji računala. Dana su neka svojstva Hammingovih kodova te je opisan postupak kodiranja i dekodiranja pomoću binarnog Hammingovog koda.

U posljednjem poglavlju navedeni su teoremi zajedno sa dokazima i metode koje

mogu poslužiti pri konstrukciji dobrih kodova iz prethodno konstruiranih kodova, a koje se dosta često koriste u praksi.

1 Osnove komunikacije

Pretpostavimo da dvije osobe pokušavaju komunicirati i dijeliti informacije. Informacije se prenose od izvora preko bučnog komunikacijskog kanala do primatelja. Najčešće možemo birati kako ćemo strukturirati informaciju na izvoru te u kojem obliku ćemo je poslati komunikacijskim kanalom, ali na ponašanje komunikacijskog kanala u većini slučajeva ne možemo utjecati. Nepouzdani kanal može imati mnoge oblike. Komunicirati možemo kroz prostor, kao na primjer, razgovarati preko bučne prostorije, ili kroz vrijeme, ako na primjer napišemo knjigu koju će netko čitati godinama kasnije. Nesigurnost kanala, u kojem god obliku bila, dopušta mogućnost da će informacija koju šaljemo biti oštećena ili iskrivljena na putu od pošiljatelja do primatelja. Tako naš razgovor preko bučne prostorije može biti ugušen bukom, a naš rukopis oštećen vremenskim prilikama.

Naravno, u mnogo slučajeva, primatelj može zatražiti da mu ponovimo informaciju koju nije razumio, na primjer ako razgovaramo, pitati nas da mu ponovimo riječ koju nije razumio. No, ovo ponavljanje i ponovo slanje informacije nije baš učinkovit način korištenja vremena. Umjesto toga, postoji bolji način, a taj je da obnovimo izvornu informaciju iz verzije koju smo primili ako nije previše korumpirana. Osnovna ideja je da dodamo redundancije u našu poruku na samom izvoru, odnosno da povećamo broj simbola poruke bez da mijenjamo količinu obavijesti, radi bolje kontrole i sigurnijeg prenošenja. Svaki jezik ima toliku prirodnu redundaciju da veliki dio poruke može biti izgubljen ili krivo prenešen, a da je rezultat i dalje razumljiv. Tako na primjer, ako sjedimo u tramvaju i kroz gužvu pročitamo: "AK OO MŽŠ PRČITTI MŽŠ SE ZAOSLITI", većina ljudi moći će razumjeti poruku iako veliki dio nje nije prenešen.

Jedan od najstarijih načina kodiranja sa svrhom nadziranja pogrešaka prilikom prenošenja poruka putem računala je dodavanje jednog kontrolnog bita nizu bitova koji predstavljaju informaciju koju želimo poslati. Taj bit nam daje informaciju je li broj jedinica u nizu koji šaljemo paran ili neparan. Pretpostavimo da šaljemo niz od 26 bitova i svaki od njih može biti 0 ili 1. Ovom nizu od 26 bitova dodamo još jedan dodatni bit kojeg odredimo na temelju prethodnih 26 i to na način će dodatni bit biti 0 ako početni niz sadrži paran broj jedinica, dok će dodatni bit biti 1 ako početni niz sadrži neparan broj jedinica. Niz od 27 bitova koji nastane na taj način uvijek će sadržavati paran broj jedinica. Dodavanjem ove male redundancije nismo značajno izmjenili sadržaj poruke koju prenosimo, od 27 bitova njih 26 pre-

nosi informaciju, ali sada smo osigurali dodatnu mogućnost za uklanjanje grešaka koje su se mogle javiti prilikom slanja. Ukoliko prilikom slanja informacije dođe do greške, onda će primljeni niz od 27 bitova imati neparan broj jedinica. Kako znamo da svaki poslani niz ima paran broj jedinica, možemo biti sigurni da je nešto pošlo po zlu prilikom prijenosa te sukladno tome reagirati, na primjer zahtjevom da se informacija ponovo pošalje. Nažalost, sposobnost uklanjanja grešaka je kod ove metode ograničena samo na to, mogućnost detekcije da je došlo do greške. Bez dodatnih informacija ne možemo znati koja je bila izvorna informacija budući da je primljeni niz sa neparnim brojem jedinica mogao nastati zbog greške na bilo kojem od 27 bitova poslanog niza. Također, treba uzeti u obzir da je moguće da se dogodilo više grešaka, a ne samo jedna. U tom slučaju, nećemo uvijek moći detektirati da se greška uopće pojavila. Ako se jave greške na dva bita (ili bilo kojem parnom broju bitova), onda će primljeni niz i dalje imati paran broj jedinica, pa grešku možda uopće ni ne primijetimo.

Postavlja se pitanje: možemo li dodati redundanciju koja će nam omogućiti ne samo da detektiramo da greška postoji, nego i da odredimo za koje je bitove vjerojatnije da su pogrešno preneseni? Odgovor je potvrdan. Ako na primjer imamo samo dvije moguće informacije koje želimo poslati, na primjer 0 ili 1, onda bi mogli svaku od njih ponoviti tri puta - 000 ili 111. Recimo da smo u tom slučaju primili poruku 101. Kako to nije jedan od uzoraka koji bi mogao biti poslan, znamo, kao i prije, da je došlo do pogreške, ali sada možemo i pretpostaviti što se dogodilo. Kako se u primljenoj poruci pojavljuju dvije jedinice i samo jedna nula, to snažno ukazuje na to da je izvorna poruka bila 111 i da se dogodila greška na srednjem bitu (u suprotnom bi izvorna poruka bila 000 i imali bi greške na dva bita). Zbog toga, možemo pretpostaviti da je izvorna poruka bila 111. Ovaj pristup "glasa većine" u dekodiranju će rezultirati točnim pretpostavkama tj. ispravljanju greški u primljenim porukama u najvećem broju slučajeva, kada se pojavila greška na samo jednom bitu.

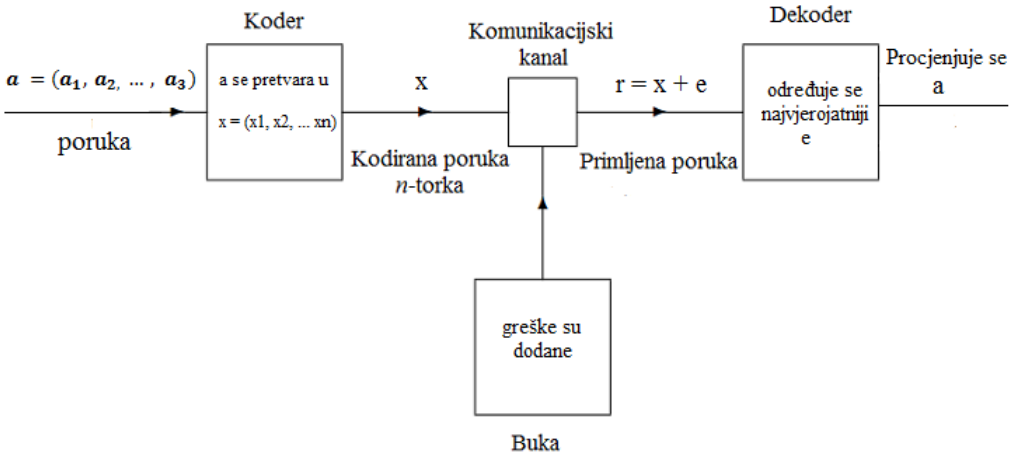
Pogledajmo sada naš kanal koji prima niz od 27 bitova. Za prenošenje bilo koje naše poruke 0 ili 1, možemo sada ponoviti poruku 27 puta. Ako ovo napravimo i onda dekodiramo koristeći "glas većine", dobit ćemo točnu poruku i ako se pojavilo čak 13 pogrešnih bitova. Ovo je svakako moćno sredstvo otklanjanja pogrešaka, ali cijenu zato plaćamo u sadržaju poruke koja se prenosi. Sada od naših 27 bitova, samo jedan nosi stvarnu informaciju, ostali su redundancija.

Imamo dakle dva različita koda duljine 27 - kod u kojem se gleda parnost broja

jedinica koji je bogat informacijama, ali ima malu sposobnost oporavka od grešaka i ponavljajući kod, u kojem se izvorna poruka ponavlja puno puta, koji je siromašan informacijama, ali se jako dobro nosi čak i sa ozbiljnim pogreškama. Potrebno je izbalansirati ovo dvoje. Dakle, treba nam metoda kodiranja koja prenosi pristojnu količinu informacija, ali se može učinkovito i oporaviti od grešaka.

Jedan od osnovnih problema kojima se bavi teorija kodiranja je pronalaženje metoda kodiranja koje omogućavaju prijenos razumne količine informacija i relativno dobro se nose sa greškama koje se mogu javiti u kodu. Postojanje ovakvih kodova je posljedica Shannonovog teorema iz 1948. (pogledati Teorem 1. koji ćemo iskazati u idućem poglavlju). Sada kada se zna da takvi kodovi postoje, glavni je cilj naći kodove tj. metode kodiranja koje će se moći primjenjivati u praksi, a koje što je bolje moguće odgovaraju zahtjevima prethodno navedenog osnovnog problema teorije kodiranja.

1.1 Opći komunikacijski sustav



Slika 1: Shannonov komunikacijski sustav

Pojednostavljeno, komunikacijski sustav sastoji se od uređene k-torke elemenata nekog skupa koja predstavlja poruku koju šaljemo. Poruka je dakle niz simbola odabranih iz nekog konačnog skupa elementarnih simbola. Tu poruku uređaj koji

se zove koder pretvara u uređenu n-torku koja predstavlja kodiranu poruku koja se onda šalje komunikacijskim kanalom. Taj postupak dodjeljivanja kodnih riječi simbolima poruke zove se kodiranje. Svaka se kodna riječ sastoji od jednog ili više simbola iz neke druge abecede. Dakle, kodiranjem se poruka (niz simbola), pretvara u niz kodnih riječi. Pod bukom podrazumijevamo greške tj. nešto što je tijekom slanja nadodano poruci u smislu da je izmijenilo kodiranu poruku. Na kraju komunikacijskog kanala uređaj koji zovemo dekoder vraća primljenu kodiranu poruku, u početnu. Ako je tijekom slanja došlo do izmjene tj. do greške, onda ju vraća u poruku koja je na neki način najbliža izvornoj poruci.

Pretpostavimo sada da hoćemo poslati poruku 1 komunikacijskim kanalom i da koder radi tako da ovu poruku kodira u 111 i nju šalje komunikacijskim kanalom. Pretpostavimo da je zbog buke poruka koju dekoder primi 101. Sada, ako dekoder radi po principu "glasa većine", budući da se dogodila samo jedna greška tijekom slanja, dekoder primljenu kodiranu poruku vraća u 111 te zaključujemo da je izvorna poruka bila 1.

Spomenimo još da smisao kodiranja ne mora nužno biti davanje poruci svojstava koja olakšavaju otkrivanje i/ili ispravljanje pogrešaka kao što smo vidjeli na primjeru ponavljajućeg koda i koda provjere parnosti. Postoji puno više razloga za kodiranje, jedan od njih može biti i pretvorba izvorne u kodiranu poruku koja je kraća od izvorne (takva vrsta kodiranja zove se kompresija) ili pretvorba u kodiranu poruku koja ima određena sigurnosna svojstva (takvo kodiranje zove se kriptografija). U radu ćemo se prvenstveno baviti zaštitnim kodiranjem koje daje poruci svojstva koja olakšavaju otkrivanje i/ili ispravljanje pogrešaka uzrokovanih smetnjama u prijenosu.

2 Shannonov teorem

2.1 Blok kodovi

Neka je Q konačan skup. *Blok kod* ili skraćeno *kod* C je bilo koji neprazni podskup skupa Q^n koji se sastoji od n-torki elemenata iz skupa Q. Dakle, možemo reći da je kod C blok kod ukoliko se kodirana informacija može podijeliti u blokove od n simbola, pri čemu se ti blokovi mogu dekodirati neovisno jedan o drugome. Broj $n = n(C)$ zovemo *duljina* koda, a skup Q^n je *prostor koda*. Broj elemenata u C je veličina koda i označava se kao $|C|$. Ako C ima duljinu n i veličinu $|C|$, kažemo da je C $(n, |C|)$ kod. Elemente prostora koda zovemo *riječi*, a one koje pripadaju skupu C, dakle, one blokove na koje možemo podijeliti naš kod, zovemo *kodne riječi*. Za skup Q kažemo da je *abeceda*.

Ako abeceda Q ima m elemenata, onda kažemo da je C *m-arni kod*. Posebno, ako je $|Q| = 2$ kažemo da je C binarni kod i onda se obično uzima $Q = \{0, 1\}$ ili $Q = \{-1, 1\}$. Ukoliko je $|C| = 1$ kažemo da je kod *trivijalan*.

Na primjer, ako je abeceda $Q = \{0, 1\}$ i $n = 3$, onda je prostor koda $Q^3 = \{000, 001, 010, 100, 011, 101, 110, 111\}$, a elementi tog skupa su riječi. Neka je $C = \{001, 101, 111, 110\}$. Onda je duljina koda $n = 3$, veličina koda $|C| = 4$, a 001, 101, 111, 110 su kodne riječi.

Definicija 1. *Ako je $x \in Q^n$, $y \in Q^n$, onda udaljenost $d(x, y)$ između x i y definiramo kao:*

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

gdje je $x = (x_1, \ldots, x_n)$ i $y = (y_1, \ldots, y_n)$.
Težinu $w(x)$ od x definiramo kao:

$$w(x) := d(x, 0).$$

Pri tome pod 0 podrazumijevamo $(0, 0, \ldots, 0)$.

Dakle, možemo reći da je težina riječi njena udaljenost do 0.

Udaljenost iz Definicije 1. koja nam za dvije n-torke daje broj pozicija na kojima se one razlikuju, zovemo još i Hammingova udaljenost i označavamo $d_H(x, y)$. Na nju trebamo gledati kao na broj grešaka potrebnih da pretvorimo x u y ili obratno. Ako koristimo komunikacijski kanal sa svojstvom da greška na poziciji i ne utječe na druge pozicije pri čemu se kao simbol u grešci može pojaviti bilo koji od preostalih $m - 1$ simbola sa jednakom vjerojatnošću, onda je, u nekim slučajevima,

Hammingova udaljenost dobar način da se detektira i ispravi greška u primljenoj poruci. Pretpostavljat ćemo da su nam sve kodne riječi jednako vjerojatne te, ako je $n_1 > n_2$, onda je uzorak greške s n_1 grešaka manje vjerojatan od onog sa n_2 grešaka. Problem s kojim se suočava primatelj kada primi kodnu riječ koja je različita od poslane je kako otkriti o kojoj se kodnoj riječi zapravo radi. Jedna strategija koja se vrlo često rabi je *princip maksimalne vjerojatnosti dekodiranja*, a to znači da kada primimo y pokušamo naći x tako da je udaljenost $d(x, y)$ minimalna. U sljedećem primjeru možemo vidjeti jednu metodu kodiranja i dekodiranja.

Primjer 1. *Pretpostavimo da komunikacijskim kanalom želimo poslati* 3*-bitnu poruku* x_1, x_2, x_3*. Uz to šaljemo i tri dodatna bita za provjeru koja definiramo kao* $x_4 := x_2 + x_3$*,* $x_5 := x_3 + x_1$ *i* $x_6 := x_1 + x_2$*. Na taj način dobijemo kod duljine šest. U ovom slučaju imamo* 8 *mogućih kodnih riječi budući da su za svaki izbor trojke* (x_1, x_2, x_3) *(koju možemo izabrati na* $2 \cdot 2 \cdot 2 = 8$ *načina), preostala tri dodatna bitna određena.*

Pretpostavimo da prenosimo c*, a primamo* $b = c + e$*. Ovaj vektor* $e = (e_1, ..., e_6)$ *zovemo uzorak greške. Vidjet ćemo da postoji jednostavan način da dekodiramo primljenu poruku što će rezultirati načinom kodiranja koji će nam u nekom smislu biti bolji od ponavljajućeg koda koji je opisan malo preciznije u prvom poglavlju. Kako je* $x_4 + x_2 + x_3 = 0$ *(jer je* $x_4 + x_2 + x_3 = 2x_2 + 2x_3$*, a ako promatramo zbrajanje u polju s dva elementa kao da se radi o binarnim znamenkama, vrijedi da je* $1 + 1$ *je* 0 *jer je* 1 *sam sebi inverz) , vidimo da je* $b_2 + b_3 + b_4 = e_4 + e_2 + e_3$*. Slične relacije vrijede i za preostala dva bita za provjeru. Definiramo:*

$$s_1 = b_2 + b_3 + b_4 = e_2 + e_3 + e_4$$

$$s_2 = b_1 + b_3 + b_5 = e_1 + e_3 + e_5$$

$$s_3 = b_1 + b_2 + b_6 = e_1 + e_2 + e_6.$$

Kako primatelj zna $b = (b_1, ..., b_6)$*, zna i* s_1, s_2, s_3*. Dakle, za danu* (s_1, s_2, s_3) *dekoder treba izabrati najvjerojatniji uzorak greške* $(e_1, ..., e_6)$ *koji daje ovu trojku, tj. koji zadovoljava gornje tri jednadžbe. Najvjerojatniji uzorak greške biti će onaj sa najmanjim brojem jedinica, tj. s najmanjim težinom.*

Ako je $(s_1, s_2, s_3) \neq (1, 1, 1)$ *onda postoji jedinstveni* e *najmanje težine koji daje* (s_1, s_2, s_3)*. Na primjer,* $(s_1, s_2, s_3) = (1, 0, 0)$ *daje* $e = (0, 0, 0, 1, 0, 0)$ *ili ako je* $(s_1, s_2, s_3) = (1, 1, 0)$ *onda je* $e = (0, 0, 1, 0, 0, 0)$*. Kada je* $(s_1, s_2, s_3) = (1, 1, 1)$ *imamo tri jednako vjerojatne mogućnosti za* e*:* $(1, 0, 0, 1, 0, 0)$, $(0, 1, 0, 0, 1, 0)$, $(0, 0, 1,$

$0, 0, 1$), *pa dekoder mora izabrati jednu od njih. Pretpostavimo da odlučimo birati i popraviti grešku s najmanjom težinom ukoliko postoji jedinstvena, a u suprotnom biramo proizvoljno. Koja je vjerojatnost da ćemo točno dekodirati poruku? Ukoliko nema grešaka ili postoji samo jedna, uvijek ćemo točno dekodirati, jer postoji jedinstveni e najmanje težine. Ukoliko postoje dvije greške, točno ćemo dekodirati samo u jednom od tri slučaja. Ako sa p označimo vjerojatnost da poslani bit ne odgovara primljenom, odnosno, da se dogodila pogreška, a stavimo da je* $q = 1 - p$, *tj. vjerojatnost da nije došlo do pogreške, onda je vjerojatnost da smo točno dekodirali sva tri simbola* x_1, x_2, x_3 *jednaka zbroju vjerojatnosti da smo točno dekodirali u sva tri slučaja: ukoliko nema greške, ako ima jedna te ako imaju dvije greške. Stoga ta vjerojatnost iznosi:*

$$P = q^6 + 6q^5p + \binom{6}{2}q^4p^2 \gg q^3,$$

pri čemu je q^3 *vjerojatnost da smo točno primili poruku* (x_1, x_2, x_3) *bez da smo ju kodirali prije slanja.*

Ovaj kod kao i ponavljajući kod nam pomažu da ispravimo pogreške koje su se dogodile prilikom slanja, ali bi htjeli vidjeti koji je od njih bolji. Jedna važna mjera za učinkovitost koda je stopa koda, koju ćemo definirati u nastavku.

U prethodnom primjeru mogli smo vidjeti da, u slučaju da su se dogodile dvije greške, nismo imali jedinstvenu kodnu riječ kojom bismo dekodirali te smo birali na slučajan način jednu od tri moguće. Sposobnost koda da otkrije ili ispravi pogrešku ovisi o najmanjoj udaljenosti svih parova kodnih riječi nekog koda C. U tu svrhu imamo sljedeću definiciju.

Definicija 2. *Najmanja udaljenost netrivijalnog koda C je*

$$\min\{d(x, y)|x \in C, y \in C, x \neq y\}.$$

Najmanja težina od C je

$$\min\{w(x)|x \in C, x \neq 0\}.$$

Tako je naprimjer najmanja udaljenost ponavljajućeg koda duljine n jednaka n (jer svaki bit koji šaljemo ponavljamo n puta, a prema definiciji promatramo samo različite kodne riječi kad određujemo najmanju udaljenost). Kod kodova sa dodatnim bitom koji provjerava parnost broja jedinica, a opisan je u uvodnom dijelu, jedna pogreška davati će riječ sa neparnim brojem jedinica, pa je najmanja udaljenost 2.

Definicija 3. *Ako je $|Q| = q$ i $C \subseteq Q^n$ onda stopu R od C definiramo kao*

$$R := \frac{\log_q |C|}{n}.$$

Stopa nam daje omjer dijela koda koji je koristan tj. koji nije redundancija i prenosi stvarnu informaciju. Tako, ako stopa koda iznosi k/n, to znači da na svakih k bitova korisne informacije koder generira ukupno n bitova od kojih su njih $n - k$ redundancija. Dakle, ponavljajući kod kojeg smo opisali u prvom poglavlju ima stopu $\frac{1}{n}$, dok kod provjere parnosti ima stopu $\frac{n-1}{n}$.

Definicija 4. *U Q^n sferu radijusa ρ sa središtem u x definiramo kao:*

$$S_\rho(x) = \{y \in Q^n | d(x, y) \leq \rho\}.$$

Dakle, sfera radijusa ρ sa središtem u x sastoji se od onih y koji bi mogli biti primljeni ukoliko se u poslanoj kodnoj riječi x javilo najviše ρ pogrešaka. Ponekada će nas zanimati koliko se najviše može razlikovati primljena riječ od najbliže kodne riječi. U tu svrhu definiramo

Definicija 5. *Ako je $C \subseteq Q^n$ onda definiramo pokrivajući radijus $\rho(C)$ od C kao*

$$\max\{\min\{d(x, c) | c \in C\} : x \in Q^n\}.$$

Pokrivajući radijus je najmanji ρ takav da sfere $S_\rho(c), c \in C$, pokrivaju cijeli skup Q^n. Ukoliko je ρ najveći cijeli broj takav da su sfere $S_\rho(c), c \in C$ disjunktne, tj. imaju prazan presjek, onda je najmanja udaljenost $d = 2\rho + 1$.

Definicija 6. *Kod $C \subseteq Q^n$ sa najmanjom udaljenosti $2e + 1$ zove se savršeni kod ako je svaki $x \in Q^n$ udaljen najviše za e od točno jedne kodne riječi.*

To znači da za svaku moguću riječ w iz Q^n postoji jedinstvena kodna riječ u C koja se razlikuje u najviše e simbola od w, tj. sve moguće kodne riječi nalaze se u točno jednoj od kugli radijusa e. Činjenica da je najmanja udaljenost jednaka $2e+1$ znači da se kod može oporaviti tj. može prepoznati i ispraviti e grešaka ukoliko se pojave, pa onda kažemo da se radi o *e-ispravljajućem kodu*. Ovo je svojstvo vrlo korisno kod dekodiranja, jer koja god kodna riječ dođe u dekoder, on će uvijek moći naći originalnu kodnu riječ koja je bila poslana. Iz toga slijedi

Sphere-packing uvjet
Neka je $|Q| = q$. *Ako je* $C \subseteq Q^n$, *savršeni e-ispravljajući kod onda je*

$$|C| \sum_{i=0}^{e} \binom{n}{i} (q-1)^i = q^n.$$

Primjer jednog savršenog koda je ponavljajući binarni kod neparne duljine n koji se sastoji od riječi 0 i 1. Neka je na primjer $n = 3$ i naš kod $C = (111, 000)$. Kako je $Q^n = \{000, 001, 010, 100, 011, 101, 110, 111\}$, očito je da se svaka moguća riječ razlikuje najviše za 1 od jedne od dviju kodnih riječi 111 i 000 te je ovo primjer 1-ispravljajućeg koda.

2.2 Shannonov teorem

Od dobrog koda želimo tri stvari:
1. Želimo veliku minimalnu udaljenost. Na primjer ako je $d = 2e + 1$ onda možemo automatski ispravljati greške težine najviše e.
2. Želimo da $|C|$ bude što je veći mogući kako bi mogli slati puno informacija.
3. Želimo dobre algoritme za kodiranje i dekodiranje.

Treći zahtjev može predstavljati poteškoću. Najčešće kada imamo poruku, nije problem odlučiti kao koju kodnu riječ ćemo ju poslati, odnosno kako ćemo je kodirati, no dekodiranje često može biti dosta teško. Ako primimo poruku iz skupa Q^n kako ćemo odrediti koja joj je kodna riječ najbliža? Ukoliko naš kod nije strukturiran onda ovo vrlo često predstavlja dosta težak problem, ali ako naš kod ima nekakvu strukturu, često to možemo iskoristiti kako bi riješili problem.

Ako uzmemo u obzir sve binarne kodove stope R, postavlja se pitanje možemo li naći onaj kod za koji je vjerojatnost da smo napravili pogrešku prilikom dekodiranja proizvoljno mala?

Pretpostavimo da koristimo kod C koji se sastoji od M riječi duljine n. Ako su $x_1, x_2, \ldots, x_M$ kodne riječi, i ako koristimo princip maksimalne vjerojatnosti dekodiranja (znači ako primimo x tražimo y tako da je $d(x, y)$ minimalno), neka je P_i vjerojatnost da smo krivo dekodirali poslani x_i. U tom slučaju je vjerojatnost pogrešnog dekodiranja primljene poruke

$$P_c := \frac{\sum_{i=1}^{M} P_i}{M}.$$

Ako sada uzmemo u obzir sve moguće kodove C sa danim parametrima, definiramo:

$$P^*(M, n, p) := \textit{minimalna vrijednost od } P_c,$$

pri čemu je p vjerojatnost da se na poslanom simbolu dogodila greška. Sa q označimo vjerojatnost da nije došlo do greške tj. $q = 1 - p$.

Teorem 1. *(Shannon, 1948.) Ako je* $0 < R < 1 + p\log_2 p + q\log_2 q$ *i* $M_n := 2^{\lfloor Rn \rfloor}$ *onda* $P^*(M_n, n, p) \to 0$ *ako* $n \to \infty$.

Ovaj teorem nam kaže da ukoliko je stopa manja od kapaciteta kanala (to je izraz $1 + p\log p + q\log q$), onda za dovoljno veliku duljinu n postoje kodovi stope R sa proizvoljno malom greškom dekodiranja. Kada je stopa veća od kapaciteta kanala, ne možemo dobiti proizvoljno malu grešku dekodiranja. Pod kapacitetom kanala podrazumijevamo najveću moguću količinu informacije po simbolu koju u prosjeku možemo prenijeti komunikacijskim kanalom uz proizvoljno malu vjerojatnost pogreške.

Dakle, Shannonov teorem nam govori da dobri kodovi postoje, ali ne govori nam kako ih konstruirati.

3 Linearni kodovi

3.1 Linearni kodovi

Prije definiranja linearnih kodova podsjetimo se najprije kako definiramo polje i vektorski prostor.

Definicija 7. *Neka je K neprazan skup na kojem su zadane dvije binarne operacije $+, \cdot$ ($+ : K \times K \to K$, $\cdot : K \times K \to K$). Uređena trojka $(K, +, \cdot)$ je polje ako vrijedi:*

i) $x + y = y + x \quad \forall x, y \in K$

ii) $x + (y + z) = (x + y) + z \quad \forall x, y, z \in K$

iii) $\exists e \in K$ *takav da je* $x + e = e + x = x \quad \forall x \in K$

iv) $\forall x \in K \ \exists -x \in K$ *takav da je* $x + (-x) = -x + x = e$

v) $x \cdot y = y \cdot x \quad \forall x, y \in K \setminus \{e\}$

vi) $x \cdot (y \cdot z) = (x \cdot y) \cdot z \quad \forall x, y, z \in K \setminus \{e\}$

vii) $\exists 1 \in K \setminus \{e\}$ *takav da je* $x \cdot 1 = 1 \cdot x = x \quad \forall x \in K \setminus \{e\}$

viii) $\forall x \in K \setminus \{e\} \ \exists \frac{1}{x} \in K \setminus \{e\}$ *takav da je* $x \cdot \frac{1}{x} = 1$

ix) $x \cdot (y + z) = x \cdot y + x \cdot z \quad x, y, z \in K.$

Definicija 8. *Neka je V neprazan skup i K polje. Neka su zadane operacije $+ : V \times V \to V$ i $\cdot : K \times V \to V$. Uređena trojka $(V, +, \cdot)$ se naziva vektorski prostor nad poljem K ako vrijedi:*

i) $x + y = y + x \quad \forall x, y \in V$

ii) $x + (y + z) = (x + y) + z \quad \forall x, y, z \in V$

iii) $\exists e \in V$ *takav da je* $x + e = e + x = x \quad \forall x \in V$

iv) $\forall x \in V \ \exists -x \in V$ *takav da je* $x + (-x) = -x + x = e$

v) $\alpha(x + y) = \alpha x + \alpha y \quad \forall x, y \in V, \ \alpha \in K$

vi) $(\alpha + \beta)x = \alpha x + \beta x \quad \forall x \in V,\ \alpha, \beta \in K$

vii) $\alpha(\beta x) = (\alpha\beta)x \quad \forall x \in V,\ \alpha, \beta \in K$

viii) za $1 \in K$ *vrijedi* $1 \cdot x = x \quad \forall x \in V.$

Kako bi definirali kodove koje možemo učinkovito kodirati i dekodirati, potrebno je dodati više stukture prostoru koda. U ovom poglavlju bavit ćemo se kodovima koji imaju algebarsku strukturu. Do sada smo skup Q uzimali kao abecedu, a podskup C od Q^n kao kod. Sada ćemo podrazumijevati da je Q polje $\mathbb{F}_q$, gdje je $q = p^r$ i p prost. Dakle, $\mathbb{F}_q$ je konačno polje sa q elemenata. Tada je Q^n n-dimenzionalni vektorski prostor tj. izomorfan je $\mathbb{F}_q^n$. Dakle, možemo reći da je *linearni kod* duljine n nad poljem $\mathbb{F}_q$ potprostor od $\mathbb{F}_q^n$. Zbog toga kažemo da su riječi prostora koda $\mathbb{F}_q^n$ vektori te kodne riječi često zovemo *kodni vektori*. Nad skupom $\mathbb{F}_q$ moguće je definirati operacije zbrajanje i množenja u aritmetici modulo q. Uočimo da, budući da je linearni kod vektorski potprostor, a time i vektorski prostor, unutar njega možemo zbrajati i oduzimati kodne riječi te na taj način dobiti kodnu riječ istog koda te također možemo množiti vektore, tj. kodne riječi skalarom i definirati bazu vektorskog potprostora. Sama činjenica da linearni kod ima bazu znatno olakšava posao definiranja nekog koda. Linearni blok kod dakle, ne moramo definirati ispisom svih kodnih riječi, nego jednostavno definicijom vektora baze.

Definicija 9. *q-arni linearni kod C je vektorski potprostor od $\mathbb{F}_q^n$. Ako je C dimenzije k onda kažemo da je C $[n, k]$ kod.*

Od sada pa nadalje koristit ćemo $[n, k, d]$ kod kao oznaku za k-dimenzionalni linearni kod duljine n s najmanjom udaljenosti d. (n, M, d) kod biti će nam bilo koji kod s duljinom riječi n, M kodnih riječi i najmanjom udaljenosti d. Vidimo da ima smisla općenitu notaciju (n, M, d) zamijeniti sa $[n, k, d]$ kod linearnih kodova jer je broj kodnih riječi kod linearnih kodova određen dimenzijom potprostora.

Ako kod C ima najmanju udaljenost $d = 2e + 1$, onda može ispraviti do e pogrešaka u primljenoj riječi. Ako je $d = 2e$ onda će uzorak greške težine e uvijek biti uočen. Traženje najmanje udaljenosti koda općenito zahtjeva uspoređivanje svakog para elemenata tj. svake dvije kodne riječi. Dakle, ako C ima M kodnih riječi, potrebno je provjeriti $\binom{M}{2}$ para kodnih riječi da bi odredili d. Međutim, to za linearne kodove nije nužno.

Propozicija 1. *U linearnom kodu najmanja udaljenost jednaka je najmanjoj težini među svim ne-nul kodnim riječima.*

Dokaz. Neka su x i y kodne riječi u kodu C. Onda je $x - y \in C$ jer je C vektorski prostor. Sada imamo $d(x, y) = d(x - y, 0)$ što je težina od $x - y$ Dakle, možemo zaključiti da je svaka udaljenost kodnih riječi težina nekog vektora tj. kodne riječi, pa je najmanja udaljenost jednaka upravo najmanjoj težini među svim ne-nul kodnim riječima čime je teorem dokazan. □

U radu ćemo prikazati dva načina za opisati linearni kod C: pomoću generirajuće matrice i matrice provjere parnosti.

Definicija 10. *Generirajuća matrica G linearnog koda C je $k \times n$ matrica u kojoj retci predstavljaju bazu vektorskog prostora C.*

Smisao uvođenja generirajuće matrice je kraćenje zapisa linearnog blok koda te pojednostavljenje operacija kodiranja i dekodiranja, što ćemo vidjeti u nastavku.

Kod C je skup svih linearnih kombinacija redova matrice G, koji zovemo prostor redaka od G. Za danu matricu G, kod C možemo dobiti tako da množimo G s lijeve strane sa svim mogućim $1 \times k$ vektorima redaka. Na taj način dobijemo sve moguće linearne kombinacije. Dakle, vrijedi $C = \{aG | a \in Q^k\}$. Reći ćemo da je G u *standardnoj formi* ako je $G = [I_k|P]$, gdje je I_k $k \times k$ jedinična matrica, a P $k \times (n - k)$ matrica.

Primjer 2. *Neka je C $[7, 4]$-kod vektorskog prostora $\mathbb{F}_2^7$ generiran redovima matrice G u standardnoj formi*

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

Kada pomnožimo G s lijeve strane sa 16 *različitih binarnih vektora redaka duljine* 4 *dobijemo* 16 *kodnih riječi. Na primjer, možemo dobiti kodne riječi:*

$$\begin{aligned} (1, 1, 0, 0)G &= (1, 1, 0, 0, 1, 0, 1) \\ (1, 0, 1, 1)G &= (1, 0, 1, 1, 1, 0, 0) \\ (0, 0, 0, 0)G &= (0, 0, 0, 0, 0, 0, 0). \end{aligned}$$

Lista svih kodnih riječi je

0000000	1101000	0110100	0011010
0001101	1000110	0100011	1010001
1111111	0010111	1001011	1100101
1110010	0111001	1011100	0101110

Primjetimo da postoji 7 *kodnih riječi težine* 3, 7 *težine* 4, 1 *težine* 7 *i* 1 *težine* 0. *Budući da je ovo linearni kod, najmanja udaljenost ovog koda je* 3 *te je stoga* 1-*ispravljajući kod.*

Kada uzmemo u obzir sposobnost oporavka od greški koda, dva koda C_1 i C_2 su jednako dobra, ako se C_2 može dobiti iz C_1 pomoću niza operacija sljedeća dva tipa: proizvoljnom permutacijom koordinatnih mjesta u svim kodnim riječima i množenjem s bilo kojim ne-nul skalarom elemenata na bilo kojoj koordinatnoj poziciji. Uočimo da tada nema razlike u Hammingovim udaljenostima između riječi kodova C_1 i C_2, a time niti razlike u njihovoj sposobnosti da otkriju i isprave eventualnu pogrešku. Stoga za takve linearne kodove kažemo da su *ekvivalentni*.

Primijetimo da nam ove transformacije odgovaraju elementarnim transformacijama stupaca pripadne generirajuće matrice. Budući da su redci generirajuće matrice kodne riječi koje čine bazu, onda se nad redcima generirajuće matrice mogu vršiti operacije zamjene redaka , množenje retka ne-nul skalarom te množenje retka s ne-nul skalarom i dodavanjem nekom drugom retku, a da nova generirajuća matrica daje isti kod. Prema ovoj definiciji ekvivalencije, elementarne transformacije stupaca i redaka generirajuće matrice linearnog koda daju matricu ekvivalentnog koda. To odgovara činjenici da elementarne transformacije redaka i stupaca matrice daju ekvivalentnu matricu. Dakle, za svaki linearni kod postoji ekvivalentan linearni kod s generirajućom matricom oblika $G = [I_k|P]$.

3.2 Kodiranje linearnim kodovima

Generirajuća matrica je matrica čiji redci odgovaraju bazi vektorskog potprostora te se stoga bilo koja kodna riječ može dobiti kao linearna kombinacija vektora baze. Kodiranje se može izvršiti na način da simboli poruke budu koeficijenti za linearnu kombinaciju vektora baze. Neka je

$$G = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{bmatrix}$$

generirajuća matrica koda C, gdje su r_i vektori baze. Neka je $m = [m_1 m_2 \dots m_k]$ bilo koja poruka. Tada poruci m odgovara točno jedna kodna riječ x koda C koja

je jednaka linearnoj kombinaciji vektora r_i i skalara m_i.

$$x = m \cdot G.$$

Na primjer, uzmimo binarni linearni kod $[5, 2, 3]$ koji ima sljedeću generirajuću matricu:

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

Neka koder generira poruku 11. Ako primjenimo prethodno opisani postupak dobijemo sljedeći kod

$$[1\ \ 1] \cdot \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix} = [1\ \ 1\ \ 1\ \ 0\ \ 0].$$

Složenost kodiranja može se smanjiti ako koristimo standardni oblik generirajuće matrice koda. Tada se kodiranje poruke m svodi na

$$m \cdot [I_k|A] = [m, m \cdot A].$$

Tako dobivena kodna riječ satoji se od dva dijela. Prvih k pozicija zauzima sama poruka, dok ostalih $(n - k)$ pozicija predstavlja umnožak $m \cdot A$. Taj dio kodne riječi zove se dio za provjeru. Ako sada pogledamo prethodni primjer s tim da sada uzimamo G u standardnoj formi (možemo ju dobiti vrlo jednostavno, tako da zamijenimo prvi i treći stupac matrice) imamo

$$[1\ \ 1] \cdot \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} = [1\ \ 1\ \ 1\ \ 0\ \ 0].$$

Ova se kodna riječ sastoji od originalne poruke i 3 dodatna simbola za provjeru.

3.3 Dekodiranje linearnog koda

Pogledajmo najprije definiciju skalarnog produkta vektora.

Definicija 11. *Neka su $x, y \in \mathbb{F}_q^n$. Na vektorskom prostoru $\mathbb{F}_q^n$ definiramo skalarni produkt*

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i \cdot y_i.$$

Jedan od načina dekodiranja linearnog koda je pomoću matrice provjere parnosti koja će biti definirana u nastavku.

Ortogonalni komplement od C, dakle, skup svih vektora koji su okomiti na svaki vektor iz C, je vektorski potprostor te stoga i linearni kod kojeg označavamo sa $C^{\perp}$.

Definicija 12. *Ako je C $[n,k]$ kod definiramo dualni kod $C^\perp$ kao*

$$C^\perp := \{y \in \mathbb{F}_q^n \mid \forall x \in C \ \langle x, y\rangle = 0\}.$$

Ako je C $[n,k]$ kod onda je $C^\perp$ $[n, n-k]$ kod. Ako je Q konačno polje, presjek potprostora C i $C^\perp$ može biti veći od $\{0\}$, a mogu čak biti i jednaki. U tom slučaju, ako je $C = C^\perp$ kažemo da je C *samo-dualan* kod. Napomenimo i to da se u Definiciji 11. zapravo radi o simetričnoj bilinearnoj formi.

Definicija 13. *Bilinearna forma na $\mathbb{F}_q^n$ je funkcija $B : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$ koja je linearna u obje koordinate, tj. takva da za sve $x, y, z \in \mathbb{F}_q^n$ i $\alpha, \beta \in \mathbb{F}_q$ vrijedi*

1) $B(\alpha x + \beta y, z) = \alpha B(x, z) + \beta B(y, z)$,

2) $B(x, \alpha y + \beta z) = \alpha B(x, y) + \beta B(x, z)$.

Dodatno, kažemo da je bilinearna forma B simetrična ako za sve $x, y \in \mathbb{F}_q^n$ vrijedi $B(x, y) = B(y, x)$.

To znači da za skalarni produkt iz Definicije 11. vrijedi da je $\langle x, y\rangle = \langle y, x\rangle$, $\langle \alpha x + \beta x', y\rangle = \alpha\langle x, y\rangle + \beta\langle x', y\rangle$ te $\langle x, \alpha y + \beta z\rangle = \alpha\langle x, y\rangle + \beta\langle x, z\rangle$.

Ako je $G = [I_k|P]$ generirajuća matrica u standardnoj formi koda C, onda je $H = [-P^T|I_{n-k}]$ generirajuća matrica koda $C^\perp$. Ovo slijedi iz činjenice da je H odgovarajuće dimenzije i ranga i da $GH^T = 0$ povlači da je skalarni umnožak svake kodne riječi aG sa svakim retkom matrice H jednak 0. Odnosno, vrijedi

$$x \in C \iff xH^T = 0. \qquad (\star)$$

U prethodnom izrazu imamo $n-k$ linearnih jednadžbi koje mora zadovoljavati svaki kod.

Ako je $y \in C^\perp$ onda jednadžbu $\langle x, y\rangle = 0$ koja vrijedi za svaki $x \in C$ zovemo jednadžbom *provjere parnosti*. H zovemo *matrica provjere parnosti* koda C. Pomoću H možemo oporaviti vektore iz C u primljenoj poruci jer moraju biti okomiti na svaki redak matrice H (bazne vektore od $C^\perp$). Kod C možemo prikazati pomoću matrice H na sljedeći način: $C = \{x \in \mathbb{F}_q^n \mid Hx^T = 0\}$.

Promotrimo postupak dekodiranja generirajućom matricom dualnog koda - matricom provjere parnosti. Neka je dan $[5, 2, 3]$ kod sa $C = \{00000, 10110, 01101, 11011\}$ i matrica G u standardnom obliku

$$G = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \end{array}\right].$$

Prema prethodnom rezultatu zaključujemo da generirajuća matrica H dualnog koda mora imati strukturu (ako gledamo aritmetiku modulo 2): $H = [A^T|I_3]$ tj.

$$H = \left[\begin{array}{cc|ccc} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

Pretpostavimo da želimo kodirati poruku [1 1]. Množenjem te poruke s generirajućom matricom G dobijemo kodnu riječ $[1\ 1] \cdot G = [1\ 1\ 0\ 1\ 1]$. Množenjem te kodne riječi s H^T dobijemo

$$[1\ 1\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0\ 0\ 0].$$

Dobili smo vektor 0 što potvrđuje da je kodna riječ ispravna.

Primijetimo da matrica H^T svojim jedinicama unutar svakog stupca (a to su retci matrice H) određuje pozicije unutar ispravne kodne riječi na kojima zbroj vrijednosti unutar aritmetike modulo 2 mora biti jednak 0. Konkretno, prvi stupac matrice H^T definira da zbroj vrijednosti simbola ispravne kodne riječi na pozicijama 1, 2 i 3 mora biti paran. Drugi stupac to definira za pozicije 1 i 4, a treći za pozicije 2 i 5.

Uzmimo sada istu kodnu riječ, ali pretpostavimo da se dogodila pogreška na drugom bitu:

$$[1\ 0\ 0\ 1\ 1] \cdot \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [1\ 0\ 1].$$

Dobili smo vektor koji nije 0, što potvrđuje da je došlo do pogreške. Zbroj vrijednosti simbola čije pozicije određuju prvi stupac (pozicije 1, 2 i 3) i treći stupac (pozicije 2 i 5) matrice H^T nije paran broj.

Dakle, u svakom retku matrice H jedinice određuju pozicije unutar ispravne kodne riječi na kojima zbroj vrijednosti simbola mora biti paran broj te zbog tog svojstva matricu H i zovemo matrica provjere parnosti.

3.4 Sindromsko dekodiranje

Sindromsko dekodiranje je metoda koja nam omogućuje da smanjimo razinu složenosti dekodiranja.

Definicija 14. *Ako je C linearni kod sa matricom provjere parnosti H onda za primljenu kodnu riječ x definiramo sindrom od x kao xH^T.*

Uočimo da iz $(\star)$ slijedi da ako kodna riječ x pripada kodu C, onda je sindrom 0. Primjetimo da, ako je kod $C = [n, k]$, onda je sindrom vektor redak dimenzije $n - k$. Sindrom je bitan faktor u dekodiranju primljenih vektora x. Budući da je C potprostor od Q^n, možemo podijeliti Q^n na klase u odnosu na C. Dva vektora x i y su u istoj klasi ako i samo ako imaju iste sindrome ($xH^T = yH^T \iff x - y \in C$). Dakle, ako je primljen vektor x koji ima uzorak greške e onda x i e imaju iste sindrome. Slijedi da za najveću vjerojatnost dekodiranja x treba izabrati vektor e najmanje težine iz klase koja sadrži x i onda dekodirati x kao $x-e$. Vektor e zovemo *predstavnikom (reprezentantom)* klase.

Za $[n, k]$ kod nad $\mathbb{F}_q$ postoji q^k kodnih riječi i q^n mogućih primljenih poruka. Pretpostavimo da je stopa visoka. Primatelj mora znati q^{n-k} predstavnika klasa koji odgovaraju svim mogućim sindromima. Sada, q^{n-k} je puno manje od q^n. Kada kod ne bi bio strukturiran, onda bi za svaku moguću primljenu riječ x morali ispisati najvjerojatniju riječ koja je poslana.

Lako se vidi da ako C ima najmanju udaljenost $d = 2e + 1$, onda je svaki uzorak greške težine manje ili jednake e jedinstveni predstavnik neke klase jer dva vektora težine manje ili jednake e imaju udaljenost najviše $2e$ pa su stoga u različitim klasama. Ako je C savršeni kod, onda nema drugih predstavnika klasa. Ako kod C ima najmanju udaljenost $2e + 1$ i svi predstavnici klasa imaju težinu najviše $e + 1$ onda kažemo da je kod C *kvazi-savršen.*

Navest ćemo jedan primjer jednostavne metode dekodiranja. Neka je C $[2k, k]$ binarni samo-dualni kod s generirajućom matricom $G = [I_k|P]$. Algoritam za dekodiranje radi ukoliko može ispraviti tri greške i ako je vjerojatnost da se pojave više od tri greške u primljenoj poruci vrlo mala. Za dani kod matrica provjere parnosti dana je sa $H = [-P^T|I_k]$. Budući da je kod C samo-dualan, G je također matrica provjere parnosti. Neka je $y = c + e$ primljeni vektor. Pišemo e kao $(e_1; e_2)$ gdje se e_1 odnosi na prvih k mjesta, a e_2 na zadnjih k mjesta. Odredimo dva sindroma

$$s^{(1)} := yH^T = e_1P + e_2,$$

$$s^{(2)} := yG^T = e_1 + e_2P^T.$$

Ako se $t \leq 3$ grešaka javi samo u prvoj ili samo u drugoj polovici y-ona, tada $e_1 = 0$ ili $e_2 = 0$ te će onda jedan od sindroma biti težine manje ili jednake 3 i odmah imamo e. Ukoliko ovo nije slučaj, onda pretpostavka da je $t \leq 3$ povlači da su e_1 ili e_2 težine 1. U obzir uzimamo $2k$ vektora $y^{(i)}$ koje dobijemo mijenjanjem $i$te koordinate od y $(1 \leq i \leq 2k)$. Za svaki od ovih vektora odredimo $s^{(1)}$ (za $i \leq k$) odnosno $s^{(2)}$ (ako je $i > k$). Ako nađemo sindrom težine najviše 2, možemo ispraviti preostale pogreške. Ako nađemo sindrom težine 3, otkrili smo 4 pogreške pod pretpostavkom da je C kod udaljenosti 8, a ako C ima udaljenost najmanje 9 možemo ispraviti ovaj uzorak od četiri greške.

Teorem 2. *Neka je H matrica provjere parnosti za $[n, k]-$kod C u $\mathbb{F}_q^n$. Onda je svaki skup od $s-1$ stupaca u H linearno nezavisan ako i samo ako je najmanja udaljenost koda C barem s.*

Dokaz. Pretpostavimo prvo da je svaki skup od $s-1$ stupaca u H linearno nezavisan nad $\mathbb{F}_q$. Neka je $c = (c_1c_2 \dots c_n)$ neka ne-nul kodna riječ i neka su $h_1, h_2, \dots, h_n$ stupci od H. Budući da je H matrica provjere parnosti, vrijedi $Hc^T = 0$. Ovo možemo zapisati u obliku:

$$Hc^T = \sum_{i=1}^{n} c_i h_i = 0.$$

Težina od c, $w(c)$, je broj ne-nul komponenti od c. Ako je $w(c) \leq s-1$, onda imamo netrivijalnu linearnu kombinaciju koje se sastoji od manje od s stupaca od H i koja je jednaka 0. Ovo nije moguće zbog pretpostavke da je svaki skup od $s-1$ ili manje stupaca od H linearno nezavisan. Zbog toga je $w(c) \geq s$ i, budući da je c proizvoljna ne-nul kodna riječ linearnog koda C slijedi da je najmanja ne-nul težina kodne riječi veća ili jednaka s. Dakle, budući da je C linearan, prema Propoziciji 1. najmanja udaljenost koda C je veća ili jednaka s.

Da bismo dokazali obrat, pretpostavimo da je najmanja udaljenost koda C barem s. Pretpostavimo da je neki skup od $t < s$ stupaca od H linearno zavisan. Bez smanjenja općenitosti, možemo pretpostaviti da su ti stupci $h_1, h_2, \dots h_t$. Tada postoje skalari λ_i u $\mathbb{F}_q$, koji nisu svi 0 i takvi da je

$$\sum_{i=1}^{t} \lambda_i h_i = 0.$$

Konstruirajmo vektor c tako da se λ_i nalazi na poziciji i, $1 \leq i \leq t$, a 0 na ostalim pozicijama. Ovako konstruiran vektor c je ne-nul vektor u C budući da je $Hc^T = 0$. No, kako je $w(c) = t < s$ došli smo do kontradikcije jer prema pretpostavci svaka ne-nul kodna riječ u C ima težinu barem s. Zaključujemo da nema $s-1$ stupaca od H koji su linearno zavisni. □

Iz teorema slijedi da linearni kod C s matricom parnosti H ima najmanju udaljenost (točno) d ako i samo ako je svaki skup od $d-1$ stupaca od H linearno nezavisan i neki skup od d stupaca je linearno zavisan. Stoga ovaj teorem možemo koristiti kod određivanja minimalne udaljenosti linearnog koda ako je dana matrica parnosti. Teorem može poslužiti i u kreiranju jedan-ispravljajućih kodova (tj. kodova sa najmanjom udaljenosti 3). Da bi konstruirali takav kod, potrebno je konstruirati matricu H tako da ne postoje dva ili više stupaca koji su linearno zavisni. Jedini način da jedan stupac bude linearno zavisan je kada su svi elementi u tom stupcu 0. Pretpostavimo da su dva ne-nul stupca h_i i h_j linearno zavisna. Onda postoje ne-nul skalari $a, b \in \mathbb{F}_q$ takvi da je

$$ah_i + bh_j = 0.$$

Ovo povlači

$$h_i = -a^{-1}bh_j,$$

što znači da su h_i i h_j skalarni višekratnici. Stoga, ako konstruiramo H tako da H ne sadrži niti jedan nul stupac i da nikoja dva stupca u H nisu skalarni višekratnici, onda će H biti matrica provjere parnosti linearnog koda čija je udaljenost barem 3.

Primjer 3. *Nad poljem $\mathbb{Z}_3$ promotrimo matricu*

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

Matrica H ne sadrži nul stupce te niti jedan stupac nije skalarni višekratnik nekog drugog stupca. Stoga možemo zaključiti da je H matrica provjere parnosti za $[5, 2]$-kod u $\mathbb{F}_3^5$ sa najmanjom udaljenosti barem 3.
Da bismo pogledali kod koji smo konstruirali, bilo bi korisno kada bi imali generirajuću matricu. Budući da je H generirajuća matrica za $C^\perp$, ako primijenimo ranije spomenuto: ako je $G = [I_k|P]$ generirajuća matrica u standardnoj formi koda C, onda je $H = [-P^T|I_{n-k}]$ generirajuća matrica koda $C^\perp$, možemo dobiti matricu

provjere parnosti za $C^{\perp}$ koja je generirajuća matrica za C. Provodimo elementarne transformacije redaka u H kako bi ju zapisali u standardnoj formi H'.

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \to H' = \begin{bmatrix} 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} = [I_3|P]$$

te je

$$P = \begin{bmatrix} 1 & 2 \\ 0 & 2 \\ 1 & 0 \end{bmatrix}$$

$$G = [-P^T|I_2] = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Sada možemo uzeti sve linearne kombinacije (nad $\mathbb{Z}_3$) redaka kako bi napisali 9 *kodnih riječi koda C:*

Kodna riječ	*Težina*
00000	0
20210	3
11001	3
10120	3
22002	3
01211	4
21121	5
12212	5
02122	4

$$G = \begin{bmatrix} 2 & 0 & 2 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Možemo vidjeti da smo uistinu generirali kod najmanje udaljenosti 3.

Na ovom primjeru možemo vidjeti kako je, kada radimo s linearnim kodovima, često korisno prijeći iz generirajuće matrice u matricu provjere parnosti i obratno.

3.5 Binarni Hammingov kod

Formalno binarni Hammingov kod (u daljenjem tekstu Hammingov kod) definiramo na sljedeći način

Definicija 15. *Neka je r pozitivan cijeli broj i neka je H matrica dimenzija $r \times (2^r - 1)$ čije stupce čine svi vektori dimenzije r različiti od* 0 *iz vektorskog prostora $\mathbb{F}_2^r$. Matrica H je matrica provjere parnosti Hammingovog koda.*

Dakle, možemo reći da je Hammingov kod bilo koji linearni blok kod čija matrica provjere parnosti H ima r redaka, a u stupcima ima sve moguće vektore dimenzije $r > 1$ osim vektora 0. Uzmimo da je $r \geq 2$. Uočimo da je Hammingov kod linearni blok kod $[2^r - 1, 2^r - 1 - r]$. Također jedno jako zanimljivo svojstvo Hammingovog koda je da on ima najmanju udaljenost 3. To proizlazi iz načina na koji smo konstruirali matricu provjere parnosti H Hammingovog koda. Naime, možemo uočiti da nikoja dva stupca u H nisu linearno zavisna, da nema nul stupaca te da svaka dva stupca u sumi daju neki treći, odnosno, da postoje tri stupca koja su linearno zavisna. Iz ovoga prema Teoremu 3. slijedi da je najmanja udaljenost Hammingovog koda točno 3. To svojstvo nam garantira da Hammingov kod, koji ima duljinu kodne riječi barem 7, zasigurno može ispraviti jednostruku pogrešku. U nastavku ćemo navesti još jedno važno svojstvo Hammingovih kodova.

Teorem 3. *Hammingovi kodovi su savršeni kodovi.*

Dokaz. Neka je C $[2^r - 1, 2^r - 1 - r]$ Hammingov kod nad $\mathbb{F}_2$. Ako je $x \in C$ onda broj elemenata sfere oko x radijusa 1 računamo:

$$|S_1(x)| = 1 + (2^r - 1) \cdot 1 = 2^r.$$

Zbog toga 2^{2^r-1-r} (što odgovara broju kodnih riječi) disjunktnih sfera radijusa 1 oko kodnih riječi od C sadrži $|C| \cdot 2^r = 2^{2^r-1-r} \cdot 2^r = 2^{2^r-1}$ riječi, tj. sve moguće riječi. Dakle, C je savršeni kod. (slijedi iz Definicije 6. i Sphere-packing uvjeta) □

3.5.1 Kodiranje pomoću binarnog Hammingovog koda

Promotrimo sljedeću matricu provjere parnosti Hammingovog $[7, 4, 3]$ koda:

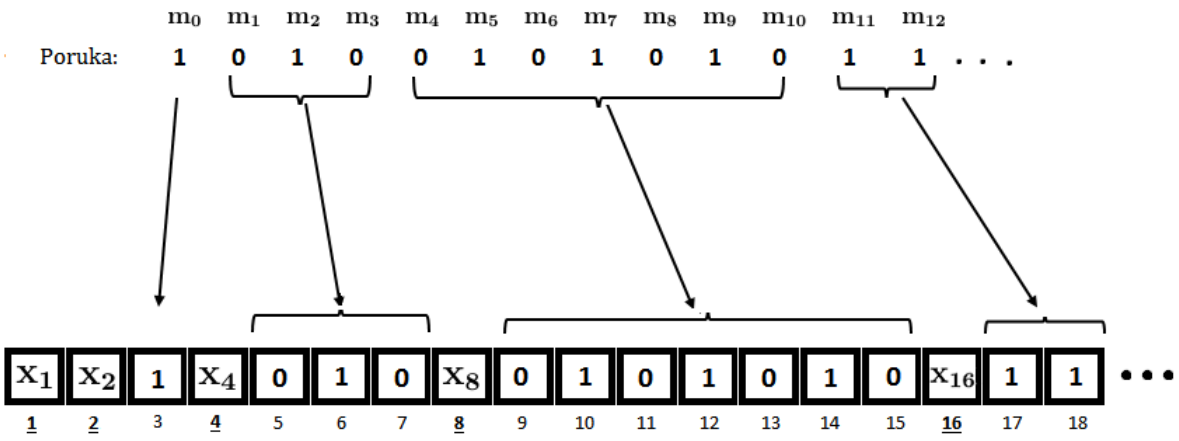
$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Primjetimo da stupci ove matrice predstavljaju binarne ekvivalente brojeva 1, 2,...7. U općem slučaju ovakvu maticu dobijemo tako da se i-ti stupac formira od bitova koji predstavljaju binarni ekvivalent broja $i = 1, \ldots 2^r - 1$, gdje je r broj redaka matrice provjere parnosti H. Matrice provjere parnosti u tom obliku najčešće se koriste u praksi, a jedan od razloga je i što omogućavaju prilično jednostavno dekodiranje što ćemo vidjeti u sljedećem potpoglavlju.

Već smo spomenuli da svaki redak matrice provjere parnosti određuje poziciju simbola kodne riječi čiji zbroj mora biti paran broj, odnosno jednak 0, u aritmetici modulo 2. Dakle, ova matrica H definira sljedeće pozicije u kodnoj riječi čiji zbroj vrijednosti simbola mora biti paran:

- prvi redak - pozicije (1), (3), (5) i (7)
- drugi redak - pozicije (2,3), (6 i 7)
- treći redak - pozicije (4, 5, 6 i 7).

Možemo uočiti da je matrica H strukturirana na način da se provjera parnosti vrši u grupama od po 1, 2 i 4 uzastopnih simbola, koje su međusobno razmaknute redom 1, 2 i 4 simbola, a prva grupa počinje na 1. 2. i 4. poziciji.
Ovakav način raspreda pozicija za provjeru parnosti je najčešći način formiranja kodne riječi Hammingovog koda. Također, u praksi se često dodatno definira da se simboli za provjeru parnosti (dodatni simboli za provjeru koje smo spominjali u potpoglavlju 3.2) postavljaju na pozicije koje odgovaraju potencijama broja 2, a simboli poruke redom između njih. Strukturu kodne riječi možemo vidjeti na Slici 2. Sa $x_1, x_2, x_8, x_{16}, \ldots$ su označene pozicije simbola za provjeru koji se određuju prema matrici provjere parnosti.



Slika 2: Standardni način formiranja kodne riječi Hammingovog koda

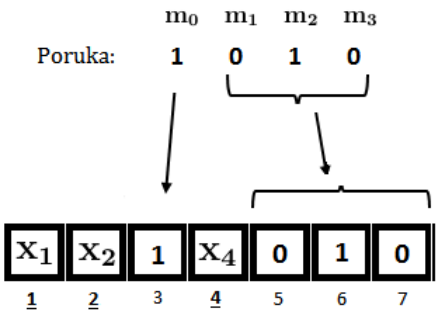
Ukoliko imamo ovakvu strukturu kodne riječi, mora vrijediti sljedeće:

$$\begin{aligned} 0 &= x_1 + x_3 + x_5 + x_7 + x_9 + x_{11} + x_{13} + \dots \\ 0 &= x_2 + x_3 \ + \ x_6 + x_7 \ + \ x_{10} + x_{11} \ + \dots \\ 0 &= x_4 + x_5 + x_6 + x_7 \ + \ x_{12} + x_{13} + x_{14} + x_{15} \ + \dots \\ &\vdots \end{aligned}$$

gdje su x_i vrijednosti simbola na poziciji i. Grupe simbola su naglašene razmakom, a vodeći signali ujedno predstavljaju signale za provjeru parnosti. Možemo ih prebaciti s lijeve strane jednakosti te na taj način dobiti izraze za njihovo izračunavanje:

$$\begin{aligned} x_1 &= x_3 + x_5 + x_7 + x_9 + \dots \\ 0 &= x_3 + x_6 + x_7 + x_{10} + x_{11} + \dots \\ 0 &= x_5 + x_6 + x_7 + x_{12} + x_{13} + x_{14} + x_{15} + \dots \\ &\vdots \end{aligned}$$

Promotrimo Sliku 3 i pogledajmo postupak formiranja jedne kodne riječi za ulaznu poruku 1010. Signal za provjeru parnosti x_1 dobiven je zbrajanjem simbola na pozicijama kodne riječi 3, 5 i 7, odnosno pozicijama 1, 2 i 4 poruke: $x_1 = 1+0+0 = 1$. Signal za provjeru parnosti x_2 dobiven je zbrajanjem signala na pozicijama kodne riječi 3, 6 i 7, odnosno pozicijama 1, 3 i 4 poruke te $x_2 = 1 + 1 + 0 = 0$. x_3 dobiven je zbrajanjem signala na pozicijama 5, 6 i 7 kodne riječi, odnosno 2, 3 i 4 poruke te je $x_3 = 0 + 0 + 1 = 1$. Na taj način dobijemo kodnu riječ 1011010.



Slika 3: Standardni način formiranja kodne riječi Hammingovog koda

Opisani postupak može se činiti kompliciranim te se zato isplati pronaći generirajuću matricu koda. Budući da matrica provjere parnosti ima 3 retka, a duljina

kodne riječi je 7, slijedi da je dimenzija koda 4. To je ujedno i broj redaka generirajuće matrice G. Broj stupaca jednak je duljini kodne riječi. Vidjeli smo da se kodna riječ Hammingovog koda formira množenjem poruke s matricom G. Prvi signal kodne riječi formira se množenjem poruke s prvim stupcem od G, drugi signal množenjem poruke s drugim stupcem, itd. U našem primjeru prvi signal formirali smo zbrajanjem 1. 2. i 4. bita poruke. Stoga će prvi stupac u G sigurno biti $[1101]^T$. drugi stupac formirali smo zbrajanjem 1., 3. i 4. signala poruke, pa je drugi stupac $[1011]^T$. Nastavljajući ovaj postupak analogno, dobijemo sljedeću generirajuću matricu:

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

Sada se kodiranje svodi ne umnožak poruke i generirajuće matrice:

$$[1010] \cdot \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} = [1011010].$$

3.5.2 Dekodiranje binarnog Hammingovog koda

Jedna dobra stvar kod Hammingovih kodova je što je dekodiranje iznimno jednstavno ukoliko se koristi struktura matrice provjere parnosti koja je prethodno opisana. Na primjer, Hammingov kod $[7, 4, 3]$ ima matricu provjere parnosti:

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

Ako primljena riječ y ima pogrešku na $i-$tom bitu, sindrom će biti binarni zapis vrijednosti i. Na primjer, ako primljena kodna riječ ima pogrešu na petom bitu, dobiva se sindrom

$$[0000100] \cdot H^T = [101],$$

što je upravo binarni zapis broja $5_{(10)} = 101_{(2)}$. Na taj način matrica provjere parnosti neposredno preslikava sindrom u indeks pozicije na kojoj se dogodila jednostruka pogreška. Dakle, ukoliko se dogodila jednostruka pogreška, postupak dekodiranja svodi se na dva koraka: izračunavanje sindroma primljene riječi y te

ako je dobiveni sindrom različit od 0, invertiranja signala na poziciji koja odgovara decimalnom ekvivalentu sindroma.

Moguće je koristiti i bilo koju drugu Hammingovu matricu provjere parnosti, ali onda se mora upotrijebiti opće procedura dekodiranja sindromom linearnih blok kodova.

Budući da je najmanja udaljenost Hammingovih kodova tri, oni mogu detektirati najviše dvije pogreške ukoliko su se dogodile ili ispraviti najviše jednu pogrešku. Zbog toga se u praksi mogu koristiti samo ondje gdje znamo da je vjerojatnost da se dogodi pogreška vrlo mala. Ovo je slučaj kod memorije računala, gdje su pogreške na bitovima iznimno rijetke te se Hammingovi kodovi vrlo često koriste.

4 Konstrukcija kodova iz drugih kodova

Mnogi dobri kodovi konstruirani su modifikacijom prethodno konstruiranih kodova. U ovom odlomku dati ćemo neke teoreme koji se koriste pri konstrukciji kodova iz drugih kodova.

Teorem 4. *Neka je C linearni $[n, k, d]$ kod nad poljem $\mathbb{F}_q$. Tada:*

i) postoji linearni $[n + r, k, d]$ kod nad $\mathbb{F}_q$ za svaki $r \geq 1$.

ii) postoji linearni $[n - r, k, d - r]$ kod nad $\mathbb{F}_q$ za svaki $1 \leq r \leq d - 1$.

iii) postoji linearni $[n, k, d - r]$ kod nad $\mathbb{F}_q$ za svaki $1 \leq r \leq d - 1$.

iv) postoji linearni $[n, k - r, d]$ kod nad $\mathbb{F}_q$ za svaki $1 \leq r \leq k - 1$.

v) postoji linearni $[n - r, k - r, d]$ kod nad $\mathbb{F}_q$ za svaki $1 \leq r \leq k - 1$.

Prije dokaza teorema spomenuti ćemo da se postupak pod *i*) zove *proširivanje koda*, a tako dobiveni kod *prošireni kod*. Obrnuti postupak, postupak pod *ii*) zovemo *probijanje koda*, a tako dobiveni kod *probijeni kod*. Uočimo da se probijanje koda svodi na uklanjanje stupaca u danom kodu te da je za dani C moguće da probijanje na različitim stupcima daje kodove koji nisu ekvivalentni. Kod koji se dobije primjenjujući postupak pod *iv*) zove se *potkod* izvornog koda.

Dokaz.

i) Definirajmo

$$C' = \{(c_1, \ldots, c_n, 0^r) | (c_1, \ldots c_n) \in C\}$$

Očito je C' linearni $[n + r, k, d]$ kod.

ii) Neka je $c \in C$ kodna riječ težine d. Neka je I skup svih indeksa r koordinata na kojima c ima vrijednosti koje nisu 0. Uklonimo sada sve koordinate u I iz svih kodnih riječi (to se postiže brisanjem stupaca u G). Kod koji dobijemo na taj način je linearni $[n - r, k]$ kod. Dimenzija je ostala nepromijenjena jer $d > r$ te stoga broj kodnih riječi ostaje isti. Nadalje $d(C') = d - r$ jer postoji kodna riječ težine $d - r$ (ne postoji kodna riječ manje težine jer bi tada $d(C) < d$).

iii) Prvo primjenimo *ii*), a zatim *i*).

iv) Neka je $\{c_1, \ldots, c_k\}$ baza od C. Neka je C' kod čija je baza $\{c_1, \ldots c_{k-r}\}$. Očito je C' linearni $[n, k - r]$ kod. Dodatno, $d(C') \geq d(C)$ jer je $C' \subseteq C$. Kako bi $d(C')$ bila točno d primjenimo postupak pod *iii*).

v) Ako je $k = n$ onda je $d = 1$ te uzimamo kod $\mathbb{F}_q^{n-r}$. Ako je $k < n$ pokazat ćemo egzistenciju linearnog $[n - r, k - r, d]$ koda za $k \geq r + 1$. Označimo matricu provjere parnosti koda C sa $H = [I_{n-k}|X]$ i obrišimo zadnjih r stupaca u H. Na taj način dobijemo matricu H_1 dimenzije $(n - k) \times (n - r)$ sa linearno nezavisnim redcima (ovo vrijedi jer X ima k stupaca i jer je $k > r$, stoga nismo obrisali niti jedan dio matrice I_{n-k} u H). Dodatno, svaki $d - 1$stupac u H_1 je linearno nezavisan kao i u H te stoga d nije smanjen. Na taj način smo dobili linearni $[n - r, k - r, d']$ kod gdje je $d' \geq d$. Ukoliko želimo reducirati d' na d možemo primjenit postupak opisan pod *iv*).

□

Teorem 5. *(Direktna suma) Neka je C_i linearni $[n_i, k_i, d_i]$ kod nad $\mathbb{F}_q$ za $i \in \{1, 2\}$. Direktna suma kodova C_1 i C_2, definirana na sljedeći način*

$$C_1 \oplus C_2 = \{(c_1, c_2) | c_1 \in C, c_2 \in C_2\}$$

je linearni $[n_1 + n_2, k_1 + k_2, min(d_1, d_2)]$ kod.

Dokaz. Lako se vidi da je $C_1 \oplus C_2$ linearni $[n_1+n_2, k_1+k_2, min(d_1, d_2)]$ kod (duljina koda slijedi direktno, a dimenzija slijedi iz činjenica da imamo $|C_1| \cdot |C_2|$ različitih kodnih riječi). Uočimo da ako je bez smanjenja općenitosti $d_1 \leq d_2$, onda je riječ najmanje težine $(c, 0)$ gdje je $c \in C_1$ gdje je $w(c) = d_1$. Odnosno, ako je $(c_1, c_2) \neq 0$ onda jedna od njih nije nul-riječ te je $w(c_1, c_2) \geq min(d_1, d_2)$. □

Nedostatak konstrukcije koda pomoću direktne sume je što se najmanja udaljenost nimalo ne poveća. Zato je u sljedećoj konstrukciji ovo poboljšano, no prije iskaza teorema navedimo jednu lemu koja će nam poslužiti u dokazu.

Lema 1. *Neka je q prost broj. Za svaki $x, y \in \mathbb{F}_q^n$ vrijedi*

$$w(x) + w(y) \geq w(x + y) \geq w(x) - w(y).$$

Teorem 6. *Neka je C_i linearni $[n, k_i, d_i]$ kod nad $\mathbb{F}_q$ za $i = 1, 2$. Onda je C definiran sa*

$$C = \{(u, u + v) | u \in C_1, v \in C_2\}$$

linearni $[2n, k_1 + k_2, min\{2d_1, d_2\}]$kod.

Dokaz. Lako se vidi da je C linearni kod duljine $2n$. Dodatno, preslikavanje $f : C_1 \oplus C_2 \to C$ definirano sa $f(c_1, c_2) = (c_1, c_1 + c_2)$ je bijekcija. Stoga je $dim(C) = dim(f(C_1, C_2)) = k_1 + k_2$ (jer imamo $|C_1| \cdot |C_2|$ kodnih riječi u C). Preostaje još pokazati da je $d(C) = min\{2d_1, d_2\}$.
Neka je $c = (c_1, c_1 + c_2) \in C$ neka ne-nul kodna riječ. Dakle, vrijedi da je $c_1 \neq 0$ ili $c_2 \neq 0$ ili oboje. Imamo dva slučaja:

$c_2 = 0$: U ovom slučaju je $c_1 \neq 0$ te

$$w(c_1, c_1 + c_2) = w(c_1, c_2) = 2w(c_1) \geq 2d_1 \geq min\{2d_1, d_2\}.$$

$c_2 \neq 0$: U ovom slučaju je

$$w(c_1, c_1+c_2) = w(c_1)+w(c_1+c_2) \geq w(c_1)+(w(c_2)-w(c_1)) = w(c_2) \geq d_2 \geq min\{2d_1, d_2\}.$$

gdje prva nejednakost vrijedi prema prethodno navedenoj lemi. Dakle, pokazali smo da je $d(C) \geq min\{2d_1, d_2\}$. Da bismo pokazali jednakost, uzmimo $c_1 \in C_1$ i $c_2 \in C_2$ takve da je $w(c_1) = d_1$ i $w(c_2) = d_2$. Nadalje, uočimo da su riječi (c_1, c_1) i $(0, c_2)$ u C te je stoga $d(C) = min\{2d_1, d_2\}$.

□

Navedeni teoremi često se koriste u praksi. Na primjer, prethodni teorem koristi se kod konstrukcije Reed-Mullerovog koda kojeg nećemo navoditi u ovom radu.

Sažetak

Današnji je svijet više nego ikad prije svijet informacija te će potreba za što učinkovitijim i pouzdanijim prenošenjem tih informacija samo rasti. Posebno važnu ulogu u tome odigrati će teorija kodiranja. U ovom radu dan je kratak uvod u ovu disciplinu. Da bi se uvidjela njezina važnost, mora se poznavati na koji način funkcioniraju uređaji koji prenose poruke na daljinu te do kakvih pogrešaka može doći prilikom prijenosa informacija. Zbog toga je u prvom poglavlju ukratko objašnjena ova problematika te je opisan opći komunikacijski sustav. Kako bi prijenos informacija bio što učinkovitiji potrebno je uskladiti količinu simbola koji u kodu služe za prijenos informacije sa onima koji služe kao redundacija i omogućavaju ispravljanje pogrešaka. Zato je potrebno konstruirati dobre kodove za koje znamo da postoje zahvaljujući Shannonovom teoremu. Kako bi definirali dovoljno učinkovite kodove potrebno je dodati više strukture prostoru koda. U radu naglasak je stavljen na linearne kodove koje možemo opisati pomoću generirajuće matrice i matrice provjere parnosti. Budući da su u računalima svi podaci prikazani u binarnom brojevnom sustavu, ne iznenađuje činjenica da se i u teoriji kodiranja posebna pozornos posvećuje binarnim kodovima, od kojih je jedan od najpoznatijih koji se dosta često zbog svojih svojstava korsti upravo u memoriji računala binarni Hammingov kod. Na kraju, budući da se mnogi dobri i pouzdani kodovi mogu se dizajnirati pomoću ranije konstruiranih kodova, u petom poglavlju opisane su metode kojima se to može postići i koje se u praksi dosta često koriste.

Ključne riječi

Komunikacijski sustav, blok kod, težina koda, najmanja udaljenost koda, linearni kod, generirajuća matrica, matrica provjere parnosti, binarni Hammingov kod, proširivanje koda, probijanje koda, direktna suma kodova

Summary

Today's world is more than ever world of information so we can expect that the need for most efficient and reliable transmission of those informations will only increase. Coding theory will play very important role in it. This work describes a brief introduction in this dicipline. If one wants to realize its importance, it is essential to know the principles of the way that work devices that transfer messages and what kind of errors can occur during transmission. In the first chapter this issue is explained and the general communication system is described. If we want the transmission of information to be more effective, it is necessary to adapt the amount of code symbols that carry information and the redundancy symbols. Because of that it is important to create good codes. The existence of such codes is a consequence of Shannon's theorem. To define effective codes it is necessary to add more structure to the code space. This work is focused on linear codes that can be described by generator matrix and parity check matrix. Since all data in computers is stored as binary patterns, it's not surprising that coding theory pay particular attention to binary codes. One of the most common binary codes that is often used is binary Hammings code. In the end, since many good codes can be created using codes that have already been constructed, few methods that are used to do this are described in chapter 5.

Key words

Communication channel, block code, code weight, minimum distance of a code, linear code, generator matrix, parity check matrix, binary Hammings code, code expanding, code puncturing, direct sum of codes

Literatura

[1] J. I. Hall, *Notes on Coding Theory*, Department of Mathematics Michigan State University, 2010.

[2] R. Hill, *A First Course in Coding Theory*, Oxford University Press Inc., New York, 2004.

[3] W. C. Huffman and V. Pless, *Fundametals of Error Correcting Codes*, Cambridge University Press, New York, 2003.

[4] J. H. van Lint, *Introduction to Coding Theory*, Springer-Verlag Berlin Heidelberg, 1992.

[5] I. S. Pandžić, A. Bažant, Ž. Ilić, Z. Vrdoljak, M. Kos, V. Sinković *Uvod u teoriju informacije i kodiranje*, Element, Zagreb, 2007.

Životopis

Zovem se Marina Lukanović, rođena sam 7.8.1993. godine u Vinkovcima gdje sam pohađala Osnovu školu Josipa Kozarca. Tijekom osnovnoškolskog obrazovanja sudjelovala sam na natjecanjima iz geografije, biologije i matematike, a u posebno lijepom sjećanju ostalo mi je natjeacanje iz vjeronauka na kojemu sam, zajedno sa svojom ekipom, osvojila 2. mjesto u državi. 2008. godine upisala sam Opću gimnaziju Matije Antuna Reljkovića u Vinkovcima. Tijekom četverogodišnjeg školovanja na mene je posebno utjecala profesorica iz matematike te moj interes za taj predmet počinje sve više i više rasti. Četiri godine zaredom sudjelovala sam na županijskm natjecanjima iz matematike, a dvije godine i na državnom, gdje sam na četvrtoj godini osvojila 9. mjesto. Maturirala sam 2012. godine i te iste godine upisala Sveučilišni nastavnički studij matematike i informatike na Odjelu za matematiku Sveučilišta J. J. Strossmayera u Osijeku. Trenutno boravim u Osijeku sa sestrom blizankom koja također studira matematiku na istom fakultetu te godinu dana mlađom sestrom koja studira fiziku na istom sveučilištu na Odjelu za fiziku. U slobodno vrijeme bavim se rukometom koji treniram već 13 godina te igram za ŽRK Vinkovci u trećoj hrvatskoj rukometnoj ligi.