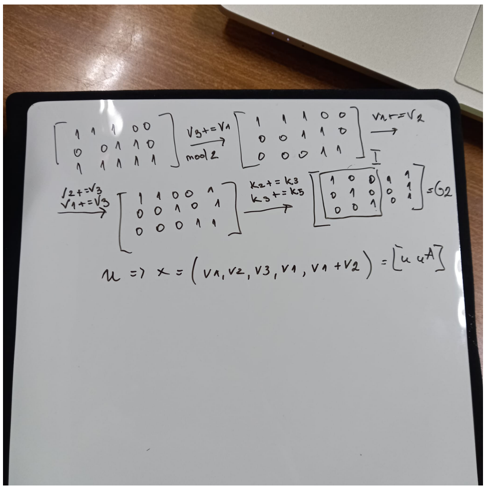
Predavanja VII

Tags	Done

Na skupom kodnih reci uvodimo strukturu - algebarska struktura. Ako su neka dva vektora kodne reci, onda su zbir i skalar takodje kodne reci. Zbir i proizvod podrazumevaju konacna polja - struktura koja odgovara racionalnim brojevima.

Konacno polje sa dva elementa, sabiranje po modulu dva - xor, a mnozenje jeste mnozenje nulom/jedinicom. Kodiranje mozemo da opisemo:

```
u = (u1,u2,u3) => u1q1 + u2q2 + u3q3 =
= (u1 + u3, u1 + u3, u1 + u2 + u3, u2 + u3, u3)
q je broj elemenata u broju
q - generatorski vektori

Kada ih slozimo u matricu dobijamo generisicu matricu koda. u
Ako na ulazu imamo k bita, onda cemo nakon kodiranja da imamo
gde je n >= k. Povecavamo broj bitova, ali sa mogucstvom dete

Kontretnu transformaciju kodiranje, originalne bitove smo zad
reci i ostatak od n-k dodavali.

Transformacijom matrice, mozemo da zadrzimo pocetni kod,
ali transofrmacijama kodiranja omogucujemo da se pojavljuju
ulazni kodovi.

Transformacija redova -> linearna transformacija,
sam kod se ne menja, ali se menja kodiranje.
Cilj je da matricu transofrmisemo promenama redosleda kolona,
ali ne menjamo originalni kod, vec koordinate u kodnoj reci
- ne gube se performase samom koda. G = [I,A]
Tim transformacijama cilj je da se tranformacija svede na G =
I - jeinicna, A - kontrolne bitove n-k

x = (u1,u2,u3,u1+u2+u3)
```



```
Rec x je kodna rec akko: x1 + x4 = 0 i x1 + x2 + x3 + x5 = 0
Svaka kodna rec zadovoljava odredjeni sistem jednacina.
Matricu H za koju Hx^t = 0 akko je x e C naziva se kontrolna
(parity-check matrica).
Vazi HG^T = 0 jer su vrste g1..g4 kodne reci pa je Hg^T = 0.
Ako je G = [I,A] onda je H = [-A^T I_n-k] -veza izmedju gener
parity check matrice.
Matrice H1..Hk su kontrolne matrice dobijene od generatorke m

Generatorska - kodiranje
Parity check - dekodiranje
```

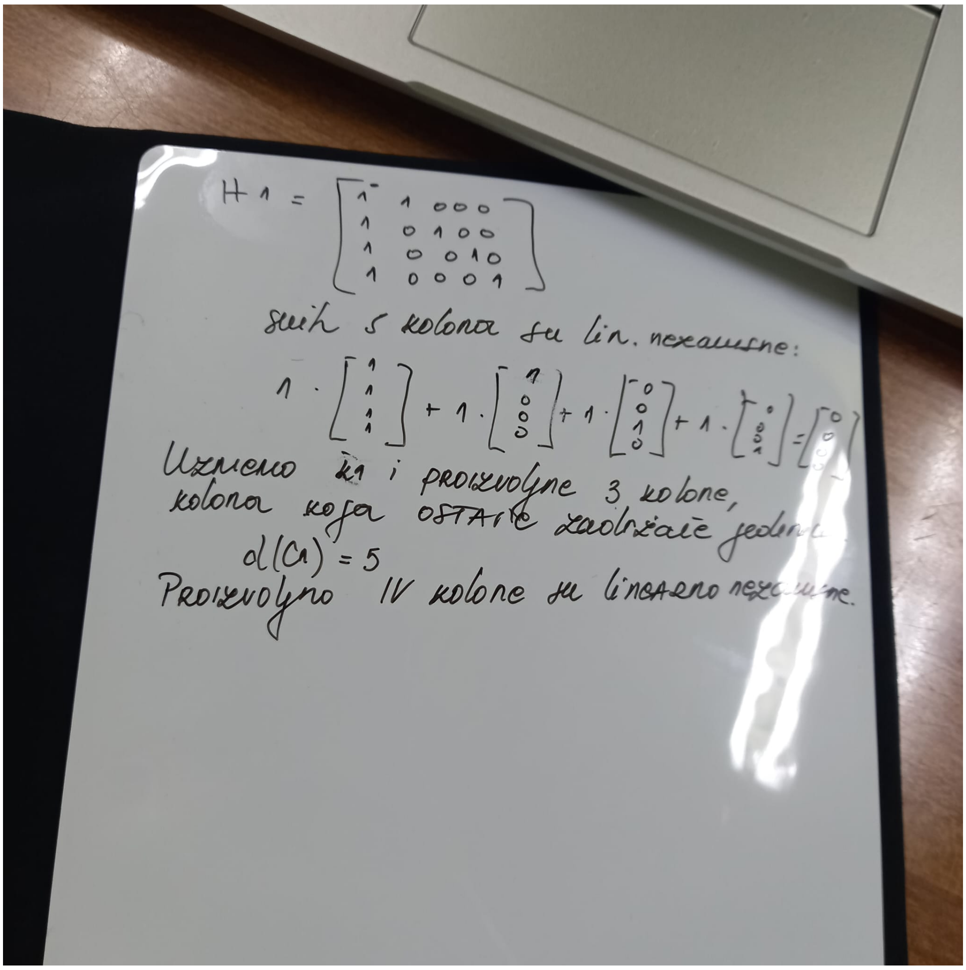
Kodno rastojanje i dekodiranje

Kodno rastojanje d(C) kod linearnih blok kodova dato je sa d(C) = min w_h(X)

Kodno rastojanje je minimalna razlika izmedju dve kodne reci X i Y. Minimalni broj bitova za koje se neke dve kodne reci razlikuju. Kod linearnog blok koda dovoljno je uzeti da je Y = 0. Vektor od svih nula je kodna rec u svakom linearnom blok koda. Ovo tvrdjenje kaze da pomocu kodnog vektora mozemo da odredimo kodno rastojanje.

```
m - minimalno rastojanje dve razlicite kodne reci
d - kodno rastojanje
Broj bitova gde se razlikuju x i y, gde se razlikuje razlika
x i y i 0. ako x i y iste vrednnosti, izlaz je 0, a u suprotn
izlaz ce biti razlictito od 0
```

```
d_h(x,y)=d_h(x-y,0) = w_h(x-y) >= m - broj ne nula pozicija u

Potrebno je da konstantujemo da je x - y element skupa C.
Kod je linearna -> x-y ulazi u minimum >= m

Sa duge strane ako uzmemo z da je proizvoljna rec:
w_h(z) = d_h(z,0) >= d
z != 0 i samim tim je to razlicito od 0.
Delta je minimum od x i y.

Drugo svojsvo je da je d(C) minimalni broj linearno nezavinis
kolona matrice H.

Kolona je lin nezavisna ako ne postoje skalari.
Vektori su linearno nezavisni ako ne postoje skalari kada
pomnozimo vektore sa sklararima od kojih je bar jedan od tih
razlicit od 0, dobijamo u zbiru 0.

Neka su b_t1^t...b_n^t kolone matice H. Posto je Hx^t = 0^t
Matrice mnoze vektore tako sto svaki vektor mnozi po jednu ko
Ukoliko imamo kodnu rec koja ima k ne nula elemenata, onda te
su ne nula i izdvojimo kolone koje odgovaraju x-evima koji su

Ako postoji kodna rec sa k ne nula elemenata, onda te kolone
zavisne - postoji k lin zavisnih kolona.
Kada uzmemo minimalni broj lin zavisnih kolona, izdvojimo ih,
Formiramo vektor x tako da na tim mestima stavimo odgovarajuc
u lin kom,a ostalo dopunimo nulama dobicemo k' ne nula elemen
Minimalan broj ne nula elmenata u kodnoj reci jednak je broju
line zavisnih kolona u matici.
```



Postupak dekodiranja pomocu parity check matrice. Pretpostavimo da je x poslata, a y primljena kodna rec koju smo dobili kroz kanal. Kada ih odizmemo dobijamo vektor e koji predstavlja gresku. Ako je e = 0 - nije doslo do greske, ako je e ≠ 0 onda je doslo do greske i tada je optimalno da primenimo LM kodiranje gde je:

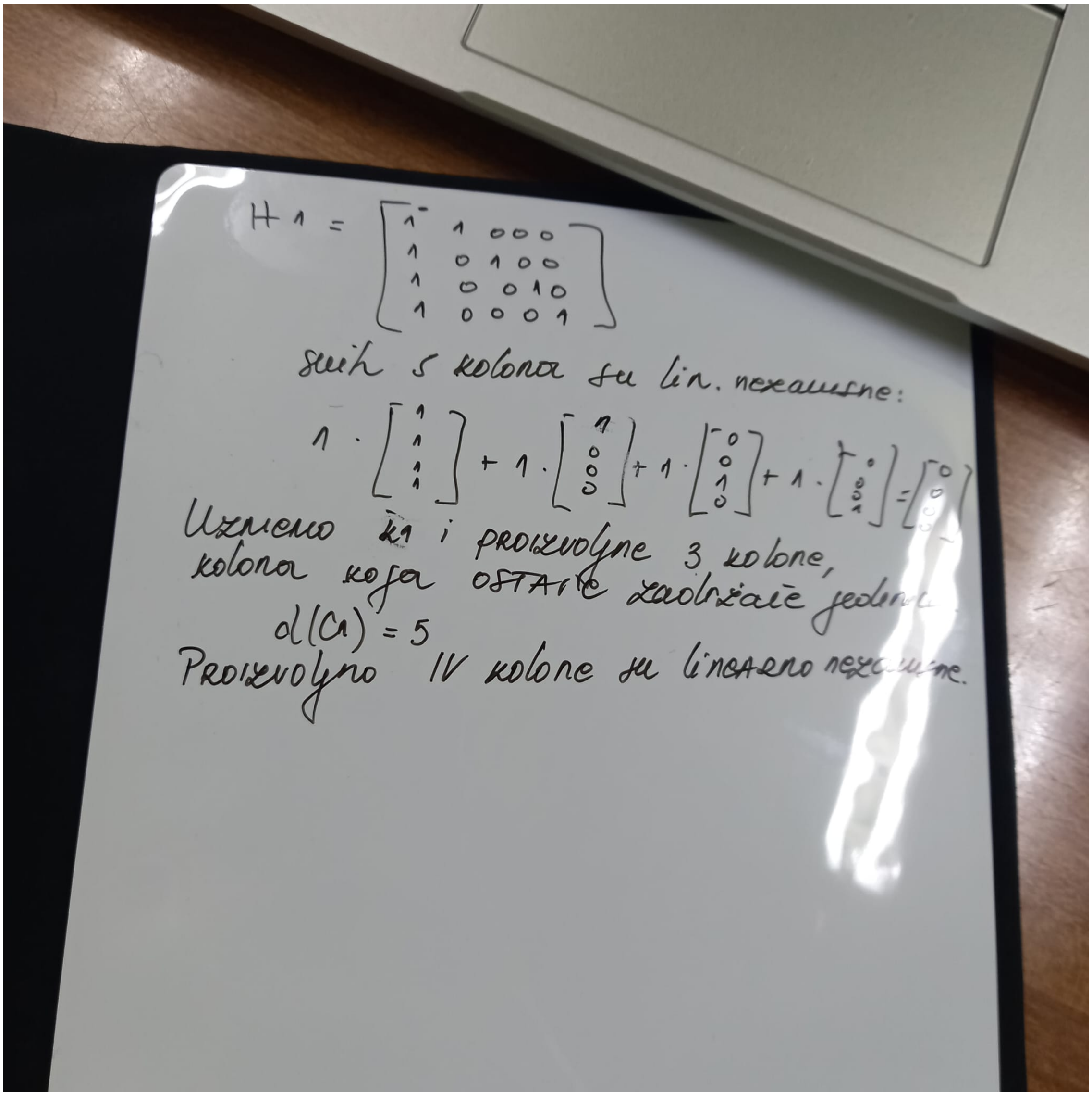
```
max p(y|x) x e C ⇒ min d_h(y,x) = w_h(y-x) x e C
```

Kodna rec x tako da je hemingova tezina vektora greske minimalna.
Trebamo da pronadjemo u procesu dekodiranja, y poznato, x nepoznato.
Trebamo da pronadjemo za koje vektor x ima minimalnu hemingovu tezinu.

```
X je kodna rec akko H_x^t = 0. Dakle s = H_y^t = H(x+e)^t = H_
```

Ml dekodiranje: Naci e tako da je H_e^t = s i da je w_h(e) minimalno i nezavisno od x.

Vektor s = H*y^t naziva se sindrom, a vektor e korektor.



Za male duzine sindroma, moguce je zapamtiti sve resenja u tabeli.

Ako usemo da zapamtimo za svako s odgovarajuci vektor e, onda je realizacija dekodera trivijalna - za to se procitamo iz memorije to e i dekodiramo.

Svodi se na elementarne aritmeticke transofmracije.
Potrebno je da nadjemo elementarnu tabelu korektora.

Imamo 3 mogucnosti za korektor za 01, performase koda se ne menja i LM dekodiranje je ispravno. Prvi put kada se pojavi neki sindrom, mi taj korektor trebamo da zabelezimo i na rakju dobijamo tabelu sindrom | korektor.

Za konkretnu realizaciju dekodera, potrebna nam je tabela sindrom | korektor.

```
dekodiranje postupak:
kada primi y izracunam s
```

```
iz tabele citam minimum i racunamo x = y-e
1. s = H_y^t
2. e = e_min(s)
3. x = y-e
```

I ako je ova procedura u osnovi jednsotavna nailazimo na 2 problema koja se odnose na tabelu, gde za q elemenata treba nam q^n elementa - eksponencijalni rast. Rezultujuca tabela q-n^k, raste takodje eksponcencijalno.

Ako uzmemo za n = 200, a k = 150, to je vec problem generisanja tabela i pamcenje tabela.

Memorija je ovde limitirana, jer ne mozemo da zapamtimo 100gb na cipu - za cuvanje tabela.
Potrebno je da ovaj problem pokusamo da resimo, dodatno pronadjemo specificne kode sa specificnim matricama G i H za koje se ova procedura dodatno pojednostavljuje.

Hammingovi kodovi

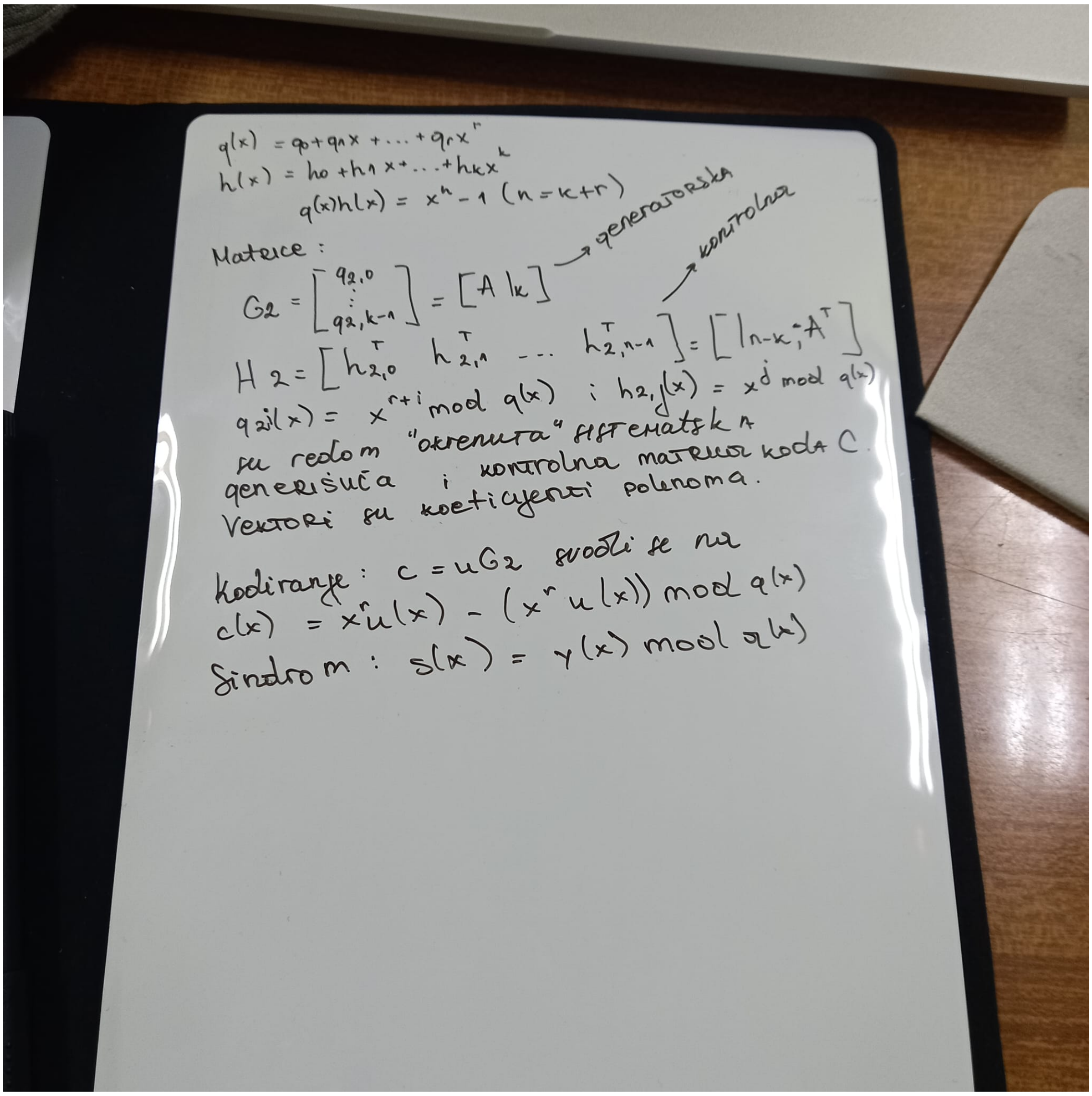
Postizemo da je broj X^n-k relativno mali da mozmeo tabelu da zapamtimo i da za njeno generisanje ne moramo da prodjemo q^n elemenata.

```
m = m-k, a da je m 2^m-1
Matica H je formata m * 2^m-1 ((n-k)*n) i sadrzi sve nenula b
vektore duzine m kao i svoje kolone.
U pitanju je (n,k) = (2^m - 1, 2^m - m -1)

Izgenerisemo sve kolone H3, permutovanjem ovih kolona
mozemo dobiti sistemacki oblik H'3 da bismo dobili odgovaraju
maticu G za ovaj kod.

Dobra stvar kod ovog koda je sto je m malo. Korektor racunamo
Ako imamo jednu gresku ei = 1, a za ej = 0 za j!=i onda imamo
maticu * vektor koji ima samo jednu jedinu, a sve ostalo nule
Samim tim kada izracunamo sindrom H3 * e, sindrom ce konkretn
biti 1 0 0 -> upravo taj sindrom ce biti binarna reprezentaci
4 i na taj nacin dobijamo indeks kodne reci gde se greska, od
jedinica nalazi na toj poziciji.

Ako je sindrom 010, korektor ce biti 01000000.
```

```
Ovaj kod ispravlja tacno jendu gresku.

Koliko god da je duzina n, kod ce ispravljati samo jednu gres
Shennova granica podrazumeva da unesemo odredjeni broj reduda
ali da n raste tako da je kolicnik k i n konacan,
da verovatnoce greske padne na 0.

Sto znaci da koliko n raste, toliko imamo vise mogucnosti
da ispravi vise gresaka gde je verovatnoca 0.

Ovaj kod je osnova za kontrukciju komplikovanijih kodova.
Konstruise se kao Hammingov kod, ali q nije 2 nego su to elme
reda 2 na neki stepen.

BCH i Read-Solomon kodovi - mogu da isprave vise od jedne gre
gde su kodiranje i dekodiranje slicni kao Hammingov kod.
```

Ciklicni kodovi

Efektivno detektovanje greska.



CRC kodovi

Kontruisu se za pogodno odabrane polinome. Glavna primena ovih kodova je detekcija gresaka. Zasto su polinomske reprezentacije dobre. Input vektor je prilicno dugacak, a parity check bitovi su kraci → check suma i nama je bitno da li je to jednako 0 ili nije. Koliko je kodno rastojanje u smislu detektovanja gresaka. Polinomne operacije se jednostavno implementiraju i realizuju na hardveru.

CRC nam govori da li je doslo do slucajnih gresaka ili ne.

LDPC kodovi - low density parity check

Za koje matrica H ima mnogo vise nula nego jedinica. Ako imamo takve kodove, pravimo bolji algoritam od sindrom algoritma, onaj koji je brzi, a dovoljno dobar u smislu ispravljanja gresaka.

Problem tabela sindroma je suvise velika.
Uz pretpostavku da je matica H retka, mozemo da uvedemo alternativne algoritme.
Za dovoljno veliko n se priblizavamo Shannoovoj granici.

```
n = 12, n -k = 9, w_r = 4, w_c = 3
w_c - je broj blokova redova u matrici H
w_r - koliko imamo blokova kolona

U prvom bloku stavljamo w_r jedinica - dijagonalno popunjavan
matrice i svakoj od sledecih grupa kolone ovih grupa
random permutujemo iz prve grupe.

Tannerov graf je grak koji dipartitan - jedna grupa cvorova s
a drugi su c-ovi. xj - cij ako je element na poziciji H_ij na
je tada 1.

X je korektna kodna rec ako je suma svih x-eva koji su u vezi
sa odgovarajucim c-om jednaki 0. H_x^t = 0.

Bit-flipping algoritam:
1. r = y
2. Svaki ci posalje vrednost w_j->i = rj svim svojim sudesima
r - trenutni vektor
3. Svi fi posalje w_i->j = sum[w_k->i] svakom svom susedu cj.
4. Svaki cj skupi sve poruke w_i->j kao i primljeni bity_k.
Ako ima vise od 0 onda r_j = 0 u suprotnom r_j = 1. Ako ima i
i jedinica, onda r_j = y_j.
5. Ponavljati korake 2-4 dok nije H_r^t = 0 ili nije dostignu
iteracija.

Ukoliko su svi c-ovi nule, znaci da nema greske i sve je ispr
U suprotnom, svaki c gde je 1, svaki x dobije povrantu inform
od c-a koja je jednaka ostalima.
x ovi salju svoje vrednosti c-ovima, c-ovi odgovaraju
gde kazu "aha po meni ti da bi bio ispunjen trebas da imas tu
vrednsot na osnovu koordinata". Kada se skupe sve vrednosti,
svaki x proceni sta je on. Ako vise c-ova kaze da treba da bu
```

```
a ne nula, x menja vrednost. To se ponavlja za odredjeni broj
iteracija dok svi c-ovi ne postanu 0 ili dok se taj broj iter
ne zavrsi i onda kazemo da je dekodiranje neuspesno.
```