**SAOBRAĆAJNI FAKULTET U BEOGRADU**

**TEORIJA INFORMACIJA I KODOVANJE**

CIKLIČNI KODOVI

1. POJAM I DEFINICIJA CIKLIČNIH KODOVA

Ciklični kodovi predstavljaju podklasu linearnih kodova i imaju veliku važnost u savremenim telekomunikacionim sistemima. Ciklični kodovi su značajni jer imaju strukturu koja se efikasno može opisati algebarski, mogu se veoma koncizno opisati - kodne reči se opisuju kao polinomi s koeficijentima iz konačnog polja, kodovanje i dekodovanje se može efikasno implementirati pomoću pomeračkih registara (ćelija za kašnjenje i elementarnih logičkih kola), većina praktično značajnih LBK su upravo ciklični.



Da bi se lakše shvatili ciklični kodovi, treba pored pojma vektora uvesti i pojam polinoma. Pri tome se biti (simboli) u kodnoj reči (vektoru) mogu posmatrati kao koeficijenti polinoma po proizvoljnoj promenljivoj x. Kao što su elementi u vektoru uređeni po svojim pozicijama, tako su sada ti elementi uređeni stepenima promenljive čiji su koeficijenti vektori. Vektoru

$$v\left(a\_{0^{ }},a\_{1^{ }},…,a\_{n-1^{ }}\right)$$

odgovara polinom (s tim što se koeficijenti, pošto se radi o polinomu, mogu pisati u proizvoljnom redosledu)

$$v\left(x\right)=v\left(a\_{n-1^{ }}x^{n-1}+…+a\_{1^{ }}x+a\_{0^{ }}\right)$$

Tako se vektor (110010) može napisati kao polinom$ x^{4}+x+1^{ }$.

Sada se posmatra skup polinoma s koeficijentima iz nekog konačnog polja. Polinomi se mogu množiti konstantom(elementom polja), sabirati i međusobno množiti.

Sabiranje polinoma je jednostavno, samo se sabiraju koeficijenti uz odgovarajuće stepene promenljive. Pošto se koeficijenti proizvoda polinoma dobijaju kao zbir proizvoda koeficijenata uz odgovarajuće stepene promenljive, to će i novodobijeni polinom imati koeficijente iz istog polja, jer je polje zatvoreno u odnosu na operacije sabiranja i množenja. Međutim, kod množenja polinoma može doći do problema. Pošto je najviši stepen proizvoda dva polinoma ravan zbiru njihovih najviših stepena, to rezultat može biti stepena većeg od *n-1*. Sada bi se sa ekvivalentnim vektorima prešlo u drugi vektorski prostor većih dimenzija. Da bi se izbegao ovaj problem može se pribeći sledećoj proceduri….

Uvođenje množenja polinoma je neophodno da bi linearna prekidačka kola mogla da množe i dele polinome, odnosno ako se kodne reči shvate kao polinomi, hardverska realizacija kodera (dekodera) može da bude veoma jednostavna.

Neka se posmatra vektorski prostor dimenzije *n-* to je ujedno i dužina kodnih vektora. Podprostor ovog prostora se naziva ciklični (kod) ako za bilo koji vektor

$$v\left(a\_{0^{ }},a\_{1^{ }},…,a\_{n-1^{ }}\right)$$

vektor
$$v\_{1^{=}}\left(a\_{n-1^{ }},a\_{0^{ }},a\_{1^{ }},…,a\_{n-2^{ }}\right)$$

dobijen cikličnom permutacijom njegovih koeficijenata takođepripada kodnom prostoru.

Ciklični potprostori su sa svoje strane podskup svih mogućih potprostora, pa su ciklični kodovi podskup linearnih kodova. Jedan potprostor je cikličan ako ga čine svi polinomi deljivi nekim polinomom *g(x)* koji je istovremeno delitelj polinoma $x^{n}+1$ . Neka je *g(x)* stepena *n-k*, tada su to linearno nezavisne klase ostataka – polinomi

$$0,g\left(x\right),xg\left(x\right),…,x^{k-1}g(x)$$

Pošto je *g(x*) stepena n-k, to će polinom $x^{k}g\left(x\right)$ biti n-tog stepena, pa će biti predstavljen svojim ostatkom po izvršenom deljenju sa $x^{n}-1$.Ti polinomi, odnosno, odgovarajući vektori, i njihove linearne kombinacije čine ciklični kod. Množenju sa x upravo odgovara cikličan pomeraj koeficijenata polinoma jer je$ $

$$x\left(a\_{0^{ }}+a\_{1^{ }}x+…+a\_{n-1^{ }}x^{n-1}\right)=a\_{n-1^{ }}+a\_{0^{ }}x+…+a\_{n-2^{ }}x^{n-1}$$

Definicija cikličnog koda u matematičkom smislu je: ***ciklični kod je ideal u algebri polinoma po modulu*** $ x^{n}-1$ ***nad poljem koeficijenata.*** Polinom *g(x)* je ***generišući polinom*** koda. Bilo koji polinom koji je delitelj $ x^{n}-1$ generiše svoj kod. Klasa ostataka (može se posmatrati i kao polinom i kao vektor) pripada kodu samo ako je deljiva sa g(x). Ako je

$$ x^{n}-1=g\left(x\right)h(x)$$

Tada i *h(x)* generiše svoj ideal- dualni kod *(n, n-k).* Polinom *h(x)* se može nazvati ***kontrolni polinom***. Ako je *g(x)* stepena *n-k*, tada je dimenzija prostora *k*, polinom *h(x)* je stepena *k* i dobija se kod *(n,k)*.

Izuzetna jednostavnost opisivanja cikličnih kodova može se shvatiti i na osnovu kratke analize:

* Za kompletan opis (n,k) blok koda potrebno je poznavanje svih $2^{k}$ kodnih reči dužine n.
* Za kompletan opis linearnog blok koda sa istim parametrima potrebno je poznavanje njegove baze – k kodnih reči dužine n (generišuća matrica).
* Za kompletan opis odgovarajućeg cikličnog koda dovoljano je znati jednu jedinu kodnu reč dužine n! Sve ostale kodne reči cikličnog koda dobijaju se kao ciklični pomeraji te kodne reči i kao linearne kombinacije ovih permutacija.

**Primer 1**

Neka je dat polinom $x^{7}+1.$ Ovde je

$x^{7}+1=\left(1+x\right)\left(1+x+x^{3}\right)(1+x^{2}+x^{3})$.

Neka je generišući polinom

*g(x)* =$ (1+x^{2}+x^{3})$.

Na ovaj način se dobija kod (7,4) jer je $n=7$, $n-k$, a ciklični kod ima dimenzije *(*$n,k$*)*.

Prva kodna reč ovog cikličnog koda se može dobiti očitavanjem koeficijenata generišućeg polinoma, a još tri linearno nezavisne kodne reči cikličnim pomeranjem ove reči:

*g(x)*=(1011000)

$x$ *g(x)*=(0101100)

$x^{2}$*g(x)*=(0010110)

$x^{3}$*g(x)*=(0001011)

Pa ovom kodu odgovara generišuća matrica

G=$\left[\begin{matrix}1&0&1&1&0&0&0\\0&1&0&1&1&0&0\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{matrix}\right]$

 Ovo su linearno nezavisni polinomi (vektori) koda (ideala). Ostale kodne reči su njihove linearne kombinacije. Kontrolni polinom je

$h\left(x\right)=\left(1+x\right)\left(1+x+x^{3}\right)= 1+x^{2}+x^{3}+x^{4}$.

U ovom slučaju je dobijen množenjem preostalih faktora, ali je isto tako mogao biti dobijen i deljenjem, tj. $h\left(x\right)=\left(x^{7}+1\right):g\left(x\right).$ I $h\left(x\right)$ obrazuje svoj kod (ideal):

$$h\left(x\right)=\left(1011100\right)$$

$$xh\left(x\right)=\left(0101110\right)$$

$$x^{2}h\left(x\right)=\left(0010111\right).$$

**Primer 2**

Za sve četvorobitne kodne reči izvršiti kodovanje kao u predhodnom primeru za generišući polinom $g\left(x\right)=\left(1+x^{ }+x^{3}\right).$

Neka je potrebno izvršiti kodovanje poruke $i=\left(1101\right)$. Proces kodovanja se može obaviti klasično, množenjem poruke odgovarajućom generišućom matricom:

$c=\left[\begin{matrix}1&1&0&1\end{matrix}\right]\left[\begin{matrix}1&1&0&1&0&0&0\\0&1&1&0&1&0&0\\0&0&1&1&0&1&0\\0&0&0&1&1&0&1\end{matrix}\right]=\left[\begin{matrix}1&0&1&0&0&0&1\end{matrix}\right]$,

Ili množenjem informacionog vektora $i\left(x\right)=x^{3}+x+1$ generišućim polinomom

$$\begin{array}{c}c\left(x\right)=i\left(x\right)g\left(x\right)=\left(x^{3}+x+1\right) \left(1+x^{ }+x^{3}\right)=x^{3}+x^{4}+x^{6}+x+x^{2}+x^{4}+1+x+x^{3}=\\ =x^{6}+x^{2}+1 \leftrightarrow (1010001)\end{array}^{ }$$

Na isti način, i za sve ostale poruke izvršili smo kodovanje i dobili:

 $i\left(x\right)\leftrightarrow c\left(x\right)$

$$\left[\begin{matrix}0&0&0&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&0&0&0&0&0&0\end{matrix}\right]$$

$$\left[\begin{matrix}0&0&0&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&0&0&1&1&0&1\end{matrix}\right]$$

$$\left[\begin{matrix}0&0&1&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&0&1&1&0&1&0\end{matrix}\right]$$

$$\left[\begin{matrix}0&0&1&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&0&1&0&1&1&1\end{matrix}\right]$$

$$\left[\begin{matrix}0&1&0&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&1&1&0&1&0&0\end{matrix}\right]$$

$$\left[\begin{matrix}0&1&0&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&1&1&1&0&0&1\end{matrix}\right]$$

$$\left[\begin{matrix}0&1&1&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&1&0&1&1&1&0\end{matrix}\right]$$

$$\left[\begin{matrix}0&1&1&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}0&1&0&0&0&1&1\end{matrix}\right]$$

$$\left[\begin{matrix}1&0&0&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&1&0&1&0&0&0\end{matrix}\right]$$

$$\left[\begin{matrix}1&0&0&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&1&0&0&1&0&1\end{matrix}\right]$$

$$\left[\begin{matrix}1&0&1&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&1&1&0&0&1&0\end{matrix}\right]$$

$$\left[\begin{matrix}1&0&1&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&1&1&1&1&1&1\end{matrix}\right]$$

$$\left[\begin{matrix}1&1&0&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&0&1&1&1&0&0\end{matrix}\right]$$

$$\left[\begin{matrix}1&1&1&0\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&0&0&0&1&1&0\end{matrix}\right]$$

$$\left[\begin{matrix}1&1&1&1\end{matrix}\right]\leftrightarrow \left[\begin{matrix}1&0&0&1&0&1&1\end{matrix}\right]$$

PROVERA CIKLIČNE RENDUNDANSE (CRC POSTUPAK)

Postupak provere ciklične redundanse (CRC - *eng. Cyclic Redundancy Check*) zasniva se na formiranju sistematskog ciklčnog koda opisanog na početku. Kod ovog postupka biti koje odgovaraju korisnim podacima prenošenim u jednom transmisionom ramu predstavljaju informacione bite koji se koduju cikličnim kodom na osnovu pravila:

$c\left(x\right)=i\left(x\right)x^{n-k}+rem\left\{\frac{i\left(x\right)x^{n-k}}{g(x)}\right\}=i\left(x\right)x^{n-k}+r(x)$,

Gde je sa$ rem\left(\frac{i\left(x\right)x^{n-k}}{g(x)}\right)$ označen ostatak pri deljenju polinoma. Na ovaj način su kontrolni i informacioni biti u kodnoj reči jasno razdvojeni, jer je u pitanju sistemski ciklični kod. Može se smatrati da je na podatke koje treba preneti prikačen CRC dodatak koji je obično mnogo manjih dimenzija od ostatka paketa $\left(n-k\ll k\right)$. Na ovaj način je uz dodavanje bita koje čine zaglavlje kompletno formiran transmisioni ram.

Pri prenosu kroz kanal, na bite koje odgovaraju ovom ramu se vrši superponiranje grešaka opisanih polinomom greške $e(x)$. Na prijemu se vrši deljenje polinoma koji odgovara primljenoj sekvenci generišućim polinomom. Ostatak pri ovom deljenju ima oblik:

$r^{'}\left(x\right)=rem\left\{\frac{c\left(x\right)+e(x)}{g(x)}\right\}$.

Pošto biti koje čine transmisioni ram odgovaraju jednoj kodnoj reči cikličnog koda, ako nije bilo grešaka pri prenosu, ostatak će biti jednak nuli. U slučaju da je ostatak različit od nule, poznato je da se desila greška pri prenosu ali ne i na kojoj poziciji. Zato CRC može da detektuje greške, ali ne i da ih ispravi. Ipak, on ima veliki značaj ako se kombinuje sa tehnikama zaštitnog kodovanja. Tada se na ovaj način omogućava provera da li je prenošeni niz informacionih bita nakon dekodovanja ispravno primlljen.

**Primer 4.**

Binarna poruka $i=(10101100)$ prenosi se kroz sistem koji koristi cikličnu proveru redundanse sa generišućim polinomom $g\left(x\right)=x^{4}+x^{3}+1$. Potrebno je odrediti kodnu reč koja odgovara poslatoj poruci.

$$i\left(x\right)=x^{5}+x^{4}+x^{2}+1 $$

$$c\left(x\right)=i\left(x\right)x^{n-k}+rem\left\{\frac{i\left(x\right)x^{n-k}}{g\left(x\right)}\right\}=\left(x^{5}+x^{4}+x^{2}+1 \right) x^{4}+rem\left\{\frac{\left(x^{5}+x^{4}+x^{2}+1 \right) x^{4}}{x^{4}+x^{3}+1}\right\}=x^{9}+x^{8}+x^{6+}x^{4}+x^{3}+x^{2}+1$$

$$\left(x^{9}+x^{8}+x^{6+}x^{4}\right):\left(x^{4}+x^{3}+1\right)=x^{5}+x^{2}+1, $$

$$r\left(x\right)= x^{3}+x^{2}+1 \rightarrow r=\left[\begin{matrix}1&0&1&1\end{matrix}\right]$$

$c=\left[\begin{matrix}1&0&1&1&1&0&1&0&1&1&0&0\end{matrix}\right]$

$$ r i$$

$\left(x^{9}+x^{8}+x^{6+}x^{4}+x^{3}+x^{2}+1\right)$:$ \left(x^{4}+x^{3}+1\right)=x^{5}+x^{2}+1 \rightarrow rem\left\{\frac{c\left(x\right)}{g(x)}\right\}=0$

Ukoliko promenimo jedan bit u primljenoj kodnoj reči (predpostavimo da je došlo do greške u prenosu):

$c=\left[\begin{matrix}1&0&1&1&1&1&1&0&1&1&0&0\end{matrix}\right]$ , imaćemo

$$ \uparrow $$

$\left(x^{9}+x^{8}+x^{6+}x^{5}+x^{4}+x^{3}+x^{2}+1\right)$:$ \left(x^{4}+x^{3}+1\right)=x^{5}+x^{2}+x$

$r\left(x\right)=x^{3}+x+1 \rightarrow rem\left\{\frac{c\left(x\right)}{g(x)}\right\}\ne 0$, ostatak neće biti jednak nuli jer je došlo do greške pri prenosu.

CRC postupak je široko primenljiv u savremenim telekomunikacijama, upravo zbog činjenice što nam govori da li je došlo do greške u prenosu. Upravo na tom principu se bazira i ARQ (Automatic Repeat Request) mehanizam za prenos podataka.

Međutim, nedostatak CRC postupka je to što se može desiti da ostatak pri deljenju polinoma bude jednak nuli, ali da se primljena kodna reč razlikuje od poslate. Upravo ovaj problem smo posmatrali, pri čemu smo u uzorku od deset, dvadeset i trideset hiljada brojali u koliko slučajeva će doći do neregularnosti.



Slika 3. Model sistema za CRC prenos.

Programi za CRC koder i dekoder su dati u prilogu. Za zadate verovatnoće greške i veličinu uzorka, dobijene su vrednosti u tabeli 2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N\p | 0.05 | 0.1 | 0.2 | 0.3 |
| 10 000 | 0.002 | 0.0044 | 0.03 | 0.0608 |
| 20 000 | 0.0004 | 0.0044 | 0.032 | 0.066 |
| 30 000 | 0.0004 | 0.0055 | 0.0291 | 0.0724 |

Tabela 2. Rezultati programa KONAČNO CRC(Prilog).

***KODER CRC***

function c=koderCRC(i)

g=[1 1 0 1]; % redosled po predavanjima g(x)=1+x+x^3

i=[0 0 0 i]; % prosirenje informacione reci i(x)\*x^(n-k)

 % u prva tri bita se smestaju redundantni biti kodne reci, a

 u preostala 4 bita su redom informacioni biti

G=fliplr(g); %redosled za matlab

I=fliplr(i);

[Q R]=deconv(I,G); % racuna se r(x)

C=abs(rem(I+R,2)); % c(x) = i(x)\*x^(n-k) + r(x)

c=fliplr(C);

***DEKODER CRC***

function [i,e]=dekoderCRC(c)

% i - informacioni biti

% e - da li je detektovana greska

i=c(4:7); % dekoder samo izdvoji zadnja 4 bita koji su informacioni

g=[1 1 0 1]; % g(x)=1+x+x^3

C=fliplr(c);

G=fliplr(g); %redosled za matlab

[Q R]=deconv(C,G);

r=fliplr(R);

s=abs(rem(r(1:3),2));

if s==[0 0 0]

 e=0; %nema greske

else

 e=1; %ima greska

end

***KONACNO CRC***

function v=konacnoCRC(N,p)

x=0; y=0; yr=0; suma=0;

x=randint(1,N);

for i=1:N/4

 I=x(4\*i-3:4\*i);

 C=koderCRC(I);

 y(7\*i-6:7\*i)=C;

end

e=verovatnocagreske(N/4\*7,p);

yr=rem(y+e,2);

for i=1:N/4

 R=yr(7\*i-6:7\*i);

 [Ir,er]=dekoderCRC(R);

 if(sum(rem(x(4\*i-3:4\*i)+Ir,2))~=0)&&(er==0)

 suma=suma+1; %broji se ako je primljena info rec Ir razlicita od

 %poslate reci x(4\*i-3:4\*i), a pri tom greska nije

 %detektovana (er==0)

 end

end

v=suma/(N/4); %odnos broja reci kod kojih nije detektovana gresaka (suma) i

 %ukupnog broja reci (N/4)

 y(7\*i-6:7\*i)=C;

end