KODET CRC NDËRTIME DHE PËRDORIME TË TYRE

AGRESA QOSJA, ARTUR BAXHAKU

Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Matematikës

e-mail: agresa.qosja1@gmail.com

Përmbledhje

Në këtë punim është paraqitur rëndësia e përdorimit të kodeve CRC. Fillimisht janë dhënë disa rezultate teorike për ndërtimin e kodeve CRC, krahasimi i tyre me kodet ciklike dhe analiza e performancës në kontrollin e gabimeve. Implementimi në software i kodeve CRC është ilustruar me anë të një shembulli. Në këtë punim është paraqitur gjithashtu edhe dobija e përdorimit të programeve kompjuterike për studimin teorik të kodeve CRC. Konkretisht, programi MATLAB përmban komanda të gatshme në teorinë e kodimit, sipas të cilave janë ndërtuar disa shembuj.

Fjalëkyçe: kodet CRC, kodet ciklike, implementimi software, MATLAB.

Abstract

This paper presents the importance of using CRC codes. Initially, there are given some theoretical results for the construction of CRC codes, to compare them with cyclic codes and the analysis of their performance in error control. The software implementation of CRC codes is illustrated by an example. In this work is also presented the necessity of using computer programming for the theoretical study of CRC codes. Specifically, the MATLAB program contains ready commands for coding, each of them constructed with some examples.

Key words: CRC codes, cyclic codes, software implementation, MATLAB.

Hyrje

Teoria e kodimit filloi të konsiderohej si pjesë e Matematikës në mesin e shekullit të XX. Për më pak se një shekull kjo fushë e Matematikës ka pasur një zhvillim të jashtëzakonshëm, duke kaluar nga përdorimi i kufizuar në institucionet shtetërore deri në përdorimin e gjerë nga ana e individëve dhe kompanive private. Ndër aplikimet më të rëndësishme përmendim sistemet bazë të të dhënave, telekomunikacion, transport, shkencat kompjuterike, etj.

Duke njohur strukturën e kodeve ciklike është përftuar një klasë e re kodesh që njihen si kode polinomiale ose ndryshe kodet ciklike të kontrollit të teprisë (Cyclic Redundany Check codes), shkurtimisht shënohen kodet CRC. Këto kode janë studiuar fillimisht nga Wesley & Peterson në vitin 1961.

Rëndësia në përdorimin e kodeve CRC lidhet drejpërdrejt me aftësinë e tyre në kontrollin e gabimeve gjatë implementimit të rrjeteve, ruajtjes dhe transmetimit të të dhënave. Kodet e kontrollit të teprisë nga vetë emërtimi janë përgjegjës për shifrat e kontrollit të një kodi, rrjedhimisht edhe për shifrat e informacionit të tij. Ndërsa fjala ciklike në emërtim lidhet me faktin se këto kode merren si shkurtim të kodeve ciklike, struktura e të cilëve është

tejet e pasur algjebrikisht, por edhe lehtësisht e implementueshme në pajisje të ndryshme. Një kod ciklik për kontrollin e teprisë emërtohet si CRC-(*n* – *k*), ku *n* – *k* është numri i shifrave të kontrollit që korrespondon edhe me shkallën e polinomit përftues të kodit. Në pjesën më të madhe të rasteve në emërtimin e kodit CRC jepen edhe inicialet e aplikimit të tij. Fusha mbi të cilën ndërtohet kodi është fusha binare $F_2 = \{0, 1\}$.

Rezultate të njohura

Ndërtimi i kodeve CRC

Le të jetë C një [*n, k*]-kod ciklik (Huffman & Pless 2003). Konsiderojmë nënbashkësinë $C_1 \subset$ C të përbërë nga të gjitha fjalët kod që kanë sasinë më të madhe të shifrave zero në *k* shifrat e tyre të informacionit në të njëjtat pozicione. Shënojmë *l* sasinë e shifrave zero. Nga secili prej elementëve të zgjedhur fshijmë *l* shifrat e zerove. Bashkësia e re e formuar është bashkësi lineare dhe ka 2^{k-l} fjalë me gjatësi *n* – *l*. Kodi i ri [*n* – *l*, *k* – *l*] është quajtur kodi i kontrollit të teprisë. Në përgjithësi emërtohet si CRC-(*n* – *k*) dhe ndryshe njihen si kode polinomiale.

Ndryshimi i kodit ciklik me kodin CRC

❖ Nëse *g*(*x*) është polinomi përftues i kodit ciklik C, atëherë mënyra e ndërtimit të kodit CRC nuk ndryshon sasinë e koeficientëve jozero të polinomit përftues.

❖ Aftësia gabim gjetëse e kodit CRC është të paktën sa aftësia gabim gjetëse e kodit ciklik nga përftohet.

❖ Kodi CRC përgjithësisht nuk është bashkësi ciklike.

❖ Procesi i kodimit dhe dekodimit i kodit CRC është e njëjtë me proceset në kodin ciklik.

Aftësia e kodit CRC në kontrollin e gabimeve

Aftësia në gjetjen dhe ndreqjen e gabimeve me anë të kodit CRC është të paktën sa aftësia e kodit ciklik në kontrollin e gabimeve (Ramabadran & Gaitonde 1988). Në praktikë ndodhin gabime rastësore me peshë një që janë lehtësisht të dallueshme nga kodi, por një rëndësi të madhe kanë gabimet me gjatësi të ndryshme të cilat e deformojnë plotësisht mesazhin.

Le të jetë *g*(*x*) polinomi përftues i kodit CRC-(*n* – k). Supozojmë se fjala e marrë pas transmetimit ose ruajtjes në sistem është *v*(*x*) = *c*(*x*) + *e*(*x*), ku *c*(*x*) është fjalë kod dhe *e*(*x*) është polinomi që paraqet formën e gabimit. Në rastin kur *v*(*x*) është shumëfish i polinomit përftues *g*(*x*), atëherë është fjalë kod. Kemi dy mundësi për polinomin *e*(*x*), është fjalë kod ose është zero. Në të kundërt polinomi *e*(*x*) është jozero dhe jo fjalë kod, rrjedhimisht *v*(*x*) nuk plotpjestohet nga *g*(*x*). Në këtë rast fjala kod pas ruajtjes në sistem ose transmetimit në kanal nuk është fjala e dërguar.

Polinomi përftues $g(x)$ i kodit CRC ka aftësinë të gjejë të gjitha gabimet me peshë një, me formë gabimi në trajtën $e(x) = x^i$, $0 \le i \le n-1$. Për më tepër nëse $g(x)$ ka si faktor polinomin $1 + x$ zotëron aftësinë e gjetjes së të gjitha gabimeve me peshë tek që mund të ndodhin në fjalët kod. Konsiderojmë format e gabimit me peshë dy me polinom të trajtës $e(x) = x^i + x^j = x^i(1 + x^{j-i})$, ku $0 \le i \le n-2$ dhe $i + 1 \le j \le n-1$. Gabimi $e(x)$ do të ishte i padiktuar nëse $g(x) \mid e(x) \Leftrightarrow g(x) \mid x^i(1 + x^{j-i})$.

Polinomi $g(x)$ nuk e ka x-in faktor dhe nëse nuk do të plotpjestonte polinomin $(1 + x^{j-i})$, atëherë $g(x) \nmid x^i(1 + x^{j-i})$. Në këtë rast do të gjendeshin të gjitha gabimet me peshë dy. Numri $m = min\{i \mid g(x) \mid (1 + x^i)\}$ quhet rend i polinomit CRC. Ndodh që kanalet e transmetimit të prodhojnë zhurma të cilat arrijnë të shkaktojnë gabime në gjatësinë e një intervali të caktuar në fjalët kod. Gabime të tilla janë quajtur gabime shpërthimi. Marrim në konsideratë rastin kur gabimi i ndodhur është në trajtën e një shpërthimi me gjatësi l dhe shënojmë me $f(l)$ probabilitetin e gabimit të padiktuar në interval. Bazuar në (Ramabadran & Gaitonde. 1988) kemi:

- $l \le n - k$ $\quad f(l) = 0,$
- $l = n - k + 1$ $\quad f(l) = \frac{1}{2^{n-k-1}},$
- $l > n - k + 1$ $\quad f(l) = \frac{1}{2^{n-k}}.$

Që këtej duket efikasiteti dhe dobia e kodeve CRC në gjetjen e gabimeve të shpërthimit. Për kode që kanë vlera të mëdha të gjatësisë së fjalëve do të kishim që probabiliteti që të diktohet gabimi i shpërthimit $1 - f(l)$ merr vlera shumë afër 1. Pra, në një pjesë të konsiderueshme të rasteve gabimet e shpërthimit kontrollohen deri në 99% të tyre.

Për shkak të performancës së lartë të kodeve CRC në kontrollin e gabimeve të transmetimit, shumë prej këtyre kodeve janë miratuar si standarde në përdorim (Koopman. 2004) CRC-5-USB, CRC-8, ATM-HEC-8, CRC-16-CCITT, CRC-32 IEEE 802.3 etj. Zakonisht në praktikë përzgjidhet një kod i standardizuar CRC me synimin që të jetë më i miri për një aplikim konkret. Realizimet në praktikë (Koopman, 2004), kanë dëshmuar se një kod i standardizuar CRC në disa raste është i papërshtatshëm për një aplikim konkret. Kjo ndodh sepse aftësia gabim-gjetëse e tij mund të mos shfrytëzohet plotësisht nga pajisje apo algoritme të ndryshme dhe si rrjedhim gjenden shumë pak gabime në krahasim me ato që ndodhin realisht. Por, mund të ndodhë që edhe kur gabimet gjenden është e pamundur që të korigjohen të gjitha (Baicheva, Dodunekov, Kazakov, 2009).

Rezultate teorike të ilustruara me shembuj

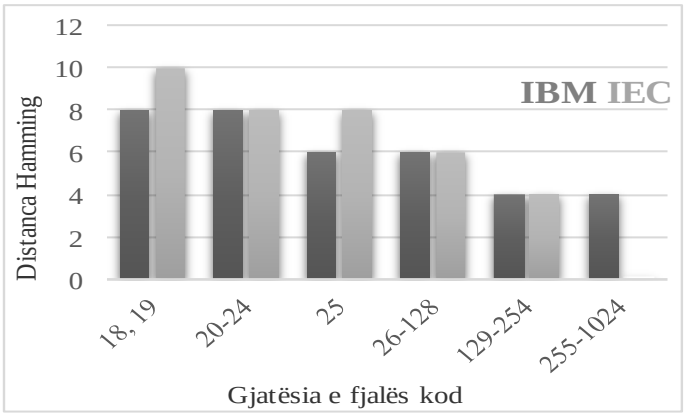
Përzgjedhja e polinomit të duhur në varësi të distancës Hamming

Marrim në konsideratë rastin e dy polinomeve të standardizuar:

IBM: $x^{16} + x^{15} + x^{13} + x^7 + x^4 + x^2 + x + 1$

IEC: $x^{16} + x^{14} + x^{12} + x^{11} + x^9 + x^8 + x^7 + x^4 + x + 1$

Në grafikun e mëposhtëm është paraqitur distanca Hamming për të dy polinomet në varësi të gjatësisë së fjalëve kod ku arrihet kjo distancë.



Në këtë rast polinomi më i mirë për përdorim përzgjidhet në varësi të gjatësisë së fjalës dhe të distancës Hamming *d* që ky polinom arrin tek kjo gjatësi. Sa më e madhe të jetë distanca Hamming aq më e lartë është performanca e kodit në kontrollin e gabimeve. Në këtë rast vërejmë se edhe midis dy polinomeve të standardizuar ka vend për përzgjedhjen e më të mirit prej tyre në gjatësi të caktuara të fjalës kod.

Implementimi në software

Në vijim do të paraqesim hapat në të cilat kalon Algoritmi Përftues për implementimin në software të kodeve CRC duke i shoqëruar me një shembull konkret. Ky algoritëm është propozuar teorikisht nga (Jain & Chouhan, 2014) për përdorim në fushat e ruajtjes dhe dërgimit të audiove dhe imazheve. Me atë të këtij algoritmi duke njohur mesazhin që do të kodohet dhe polinomin përftues të kodit, gjenden bitet e kontrollit që i korespondojnë mesazhit. Mëposhtë është paraqitur një shembull i zbatuar i këtij algoritmi.

Informacioni që do të kodohet është: $u(x) = 1 + x^2 + x^4 + x^5$. Polinomi përftues i kodit CRC-16 është $g(x) = 1 + x^2 + x^{15} + x^{16}$.

1. Futet sasia e biteve të informacionit dhe të kontrollit $k - 1 = 5$ dhe $n - k = 16$.
2. Ndërtohen vektorët me këto përmasa dhe fillojnë të mbushen me të dhëna:

$u =$

1	0	.	.	.															

g =

1	0	.	.	.											

3. Merren të gjitha të dhënat.
4. Pas plotësimit të vektorëve, rregjistrohen.
5. Rezultatet janë:

u =

1	0	1	0	1	1	0	0	0	.	.	.							.	.	0

g =

1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

6. Kontrolli:

Nëse biti i parë = 0, atëherë dil nga procesi. Në të kundërt realizohet veprimi XOR me elementët e vektorit të polinomit përftues dhe vektorit të mesazhit. Të gjitha të dhënat ruhen në një vektor ndihmës.

1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Biti i parë është 1, atëherë vendos në koordinatën e parë të vektorit ndihmës shumën e tij me bitin e parë të polinomit përftues. Vazhdon procedura me mbledhjen (komanda XOR) e biteve të tjerë të vektorëve, pavarësisht nga vlera e tyre.

1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

XOR

1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1

0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Në të njëjtën mënyrë, në pozicionin e gjetjes së një elementi njësh në vektorin ndihmës vazhdon realizimi i komandës XOR me elementët e polinomit përftues.

7. Gjenden bitet e kontrollit që i korespondojnë vektorit të mesazhit.

0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1

Fundi algoritmit. Vektori i fjalës kod është:

1	0	1	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	0	1

Rezultatin mund ta shkruajmë ndryshe: $v(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^7 + x^{15} + x^{16} + x^{19} + x^{21}$.

Përdorimi i programit MATLAB në teorinë e kodimit (Jiang, 2010)

Studimi teorik i kodeve CRC është i pamundur të realizohet pa përdorimin e programeve kompjuterike. Përmasat e të dhënave në praktikë janë tepër të mëdha, por duhet marrë në konsideratë edhe fakti që polinomet përftues janë të shkallëve të ndryshme. Duke qenë se ndaj një mesazhi me gjatësi të caktuar testohen disa polinome për aftësinë e tyre në kontrollin e gabimeve bëhet akoma më i domosdoshëm përdorimi i programeve kompjuterike. Programi MATLAB ka komanda të gatshme për zbatim në teorinë e kodimit. Në vijim po paraqesim disa shembuj të aplikuar në MATLAB.

Shembulli i parë në këtë program ka një rëndësi të madhe teorike. Studimi i ekzistencës së kodeve të përmasave të çfarëdoshme me elementë nga fusha të ndryshme është themelor në ndërtimin e kodeve në praktikë.

- Mosekzistenca e kodit [21, 5, 14] mbi F_2.

```
>> n = 21; k = 5;
>>genpoly = cyclpoly (n, k, 'all')
```

g_0	g_1	.	.	.	.	.	.	g_{15}	g_{16}
1 0 0 0 1 1 0 0 1 0 1 0 1 1 1 1 1									
1 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 1									

```
>> d = gfweight([1 1 1 1 1 0 1 0 1 0 0 1 1 0 0 0 1], 21)
    d = 10
```

Duke përdorur komandat e nevojshme në MATLAB rezulton se nuk mund të gjendet një kod me gjatësi të fjalëve 21 dhe 5 bite kontrolli mbi fushën e numrave binar.

Rëndësia e shembujve të mëposhtëm lidhet drejtëpërdrejt me eficiencën e përdorimit të programeve kompjuterike në shmangien e veprimeve llogaritëse gjatë proceseve të kodimit dhe të dekodimit. Gjithashtu edhe në gjetjen e matricës përftuese dhe të kontrollit.

- Procesi i kodimit

```
>> n = 7; k = 3;
>> m = [1 0 0];
>> c = encode( m, n, k, 'cyclic', [1 0 1 1 1])
    c = 1   0   1   1   1   0   0
```

- Procesi i dekodimit

```
>> r = [1 1 0 1  1 0];  g = [1 0 1 1 1];
>> m = decode(r, n, k, 'cyclic', g)
   m = 1   1   0
```

- Matrica përftuese dhe matrica e kontrollit

```
>> n = 7; k = 3;
>> g = [1 0 1 1 1];
>> [H, G] = cyclgen(n, g, 'nonsys')
    H =   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

    G =   1   0   1   1   1   0   0
          0   1   0   1   1   1   0
          0   0   1   0   1   1   1
```

Koment

Në këtë material është paraqitur rëndësia e kodeve CRC në përdorimin e tyre. Fillimisht është dhënë ndërtimi i tyre, ndryshimi me kodin ciklik dhe më tej analiza në kontrollin e gabimeve. Rëndësi i është kushtuar vlerësimit të dy polinomeve të miratuar si standarde duke i krahasuar midis tyre. Nga rezultatet e dhëna vërejmë se edhe midis polinomeve standarde është e mundur të përzgjidhet më i miri ndërmjet tyre. Për më tepër janë studiuar polinome të tjerë që kanë performancë më të lartë se polinomet e standardizuar (Koopman, 2004). Për zbatimin e teorisë së kodimit në aplikacione që janë duke u krijuar vazhdimisht propozojmë që është e arsyeshme të shqyrtohen polinomet e njohura si standard. Më saktë përcaktimi i tyre si standard të bëhët në varësi të gjatësive të fjalës kod, distancës Hamming dhe probabilitetit të gabimit të padiktuar. Problematikat e deritanishme lidhen me faktin se standardizimi i polinomeve përftues është bërë në një fushë shumë të gjërë, pra sipas shkallës së tij.

Është e qartë se edhe në grafikun e mësipërm pavarësisht se janë dy polinome standarde të së njëjtës shkallë ka vend për përcaktimin e "më të mirit" prej tyre në varësi të kushteve të përcaktuara. Në vijim të punimit me anë të një shembulli konkret është paraqitur implementimi në software i kodeve CRC. Në këtë mënyrë, duke vërejtur lehtësinë praktike të këtij algoritmi theksojmë së është më vend përdorimi i tij në fushat e propozuara teorikisht. Duke u nisur nga fakti se në praktikë përmasat e të dhënave janë tepër të mëdha, është e domosdoshme përdorimi i programeve kompjuterike, në këtë rast programi MATLAB. Ky program përmban një sasi të konsiderueshme të komandave të nevojshme, por nuk mund të pohojmë se është më i miri në kushtet kur ka edhe shumë programe të tjera në përdorim. Në këtë punin është parë e arsyeshme përdorimi i këtij programi sepse në varësi të komandave të gatshme që ky program përmban jep rezultate të vlefshme në përpunimin e të dhënave, por gjithashtu edhe në studimin e

njohurive teorike përpara se të ketë përpjekje në zbatimin praktik të tyre. Për shkak të thjeshtësisë në përdorimin e programit MATLAB sugjerojmë aplikimin e tij nga të gjithë të interesuarit në këtë fushë. Ky sugjerim nuk vlen vetëm për dashamirësit e këtij drejtimi, por edhe për të gjithë studiuesit që në të ardhmen mund të paraqesin rezultate shumë të rëndësishme.

Përfundime dhe drejtime për punë të mëtejshme

Gjetja e kodeve CRC më të mirë është një problem i rëndësishëm në ditët e sotme i cili ende nuk duket se ka një zgjidhje të plotë. Kjo për arsyen e thjeshtë se për çdo gjatësi të mundshme mesazhi duhet përcaktuar kodi më i mirë CRC. Një tjetër fakt është se duhet përcaktuar në fillim analiza e plotë teorike dhe më tej realizimi i të gjitha veprimeve llogaritëse sipas programeve kompjuterike më të mira. Pasi të jenë përzgjedhur teorikisht, polinomet përftues duhet që të testohen në aplikime konkrete. Por, performanca e kodeve CRC në kontrollin e gabimeve në disa raste nuk është më e mira e mundshme. Për mesazhe me gjatësi më shumë se 8000 bit të cilat gjenden në rrjete kompjuterike apo në sisteme për ruajtjen e të dhënave ky problem mbetet i hapur.

Zëvendësimi i standardeve aktuale me polinome të tjerë është i pamundur nga pikëpamja ekonomike, por rezultatet e arritura mbeten interesante për aplikimet e reja të cilat janë duke u krijuar vazhdimisht. Prandaj, studimi i të gjithë polinomeve me një shkallë të caktuar do të ndihmojë praktikuesit në përzgjedhjen e polinomit që arrin eficiencë të lartë në përdorim. Deri tani janë studiuar kodet CRC që ofrojnë nga 3 deri në 16 bite kontrolli ndaj mesazhit, gjithashtu edhe kodet me polinom përftues me shkallë 32. Ndërkohë që në praktikë janë përdorur kode CRC që ofrojnë 64 bite kontrolli. Nga këto rezultate vërejmë se mbetet akoma shumë punë për t'u bërë deri në studimin e hollësishën të të gjithë polinomeve përftues me shkallë deri në 64. Për më tepër janë propozuar polinome që zotërojnë aftësi më të mira në kontrollin e gabimeve se sa polinomet e vlerësuar si standard.

Një problem tjetër interesat në fushën e kërkimeve të ardhshme është zhvillimi i udhëzimeve për zgjedhjen e polinomit përftues më të mirë. Por, përcaktimi i të gjitha karakteristikave të nevojshme për një polinom CRC si distanca minimale, rendi i polinomit apo shpërndarja e peshave, kërkon një punë afatgjatë e cila bëhet gjithmonë e më shumë e komplikuar me rritjen e shkallës së polinomit. Nga studiuesit (Koopman, 2004) është pohuar se prezantimi i një liste të plotë polinomesh me shkallë të lartë është një sfidë vërtet e madhe dhe mbetet për t'u studiar në vitet e ardhshme.

Literatura

Huffman C. W., Pless V. (2003): Fundamentals of Error Correcting Codes, Cambridge University Press: 121 – 167

Ramabadran V. T., Gaitonde S. S. (1988): A Tutorial on CRC Computations, IEEE MICRO

Baicheva S. T., Dodunekov S., Kazakov P. (2009): Undetected error probability performance of cyclic redundancy-check codes of 16-bit redundancy, IEE Proc – Commun, Vol 147, No. 5

Koopman P. (2004): Cyclic Redundancy Code (CRC) Polinomial Selection For Embedded Networks

Jain S., Chouhan S. S. (2014): Cyclic Redundancy Code: Study and Implemmentation, IJETAE

Jiang Y. (2010): A Practical Guide to Error – Control Coding Using MATLAB, Artech House