Laboratorij za strojni vid,
Fakulteta za elektrotehniko, Univerza v Ljubljani

Komunikacije v Avtomatiki
Avditorne vaje

Matej Kristan in Janez Perš
<janez.pers@fe.uni-lj.si>

Univerza v Ljubljani

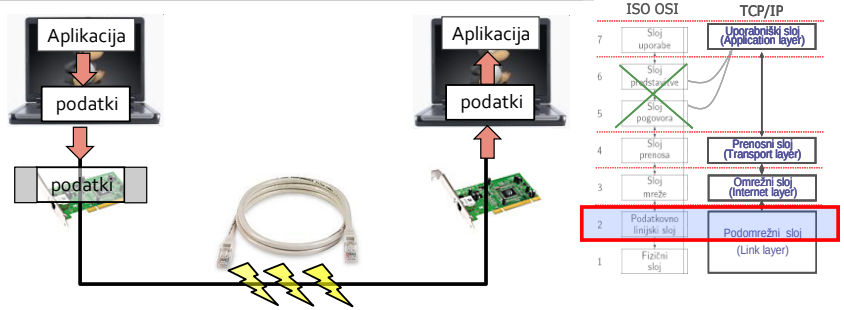
Vsebina vaj:

1. Postopki cikličnega kodiranja (CRC).
2. Postopki zgoščevanja podatkov (Huffman).

Postopki cikličnega kodiranja (CRC)

1. Zagotavljanje pravilnosti prenosa podatkov.
2. Uvod v CRC.
3. Modulo 2 aritmetika.
4. Klasični CRC.
5. (CRC s tabelo ostankov.)
6. Naloge se navezujejo na (Kovačič 2001, pp. 95-106)

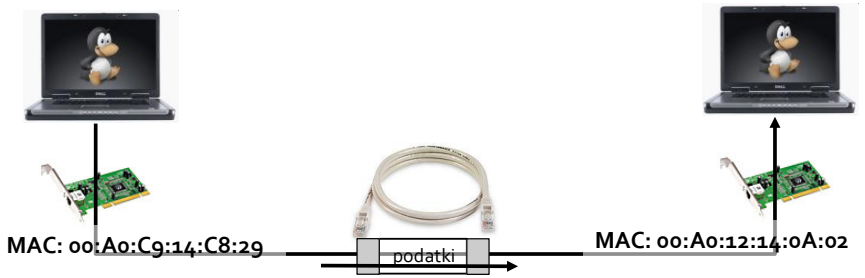
Zagotavljanje pravilnosti prenosa



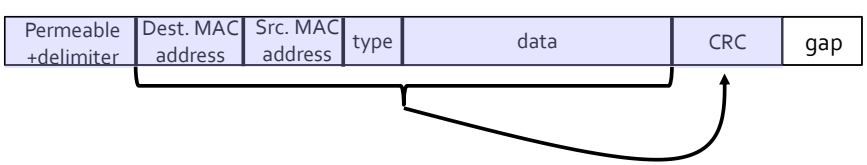
- Naprave komunicirajo preko šumnega kanala.
- Vprašanja:
  - Kako ugotoviti **kdaj je** pri prenosu okvirja **prišlo do napake**?
  - Ali je **možno popraviti napako**, če slednjo detektiramo?

1. Kontrolna vsota (checksum)
2. Hammingov kod.
3. Metoda cikličnega kodiranja (angl. **C**yclic **R**edundancy **C**heck, **CRC**).

Zagotavljanje pravilnosti prenosa



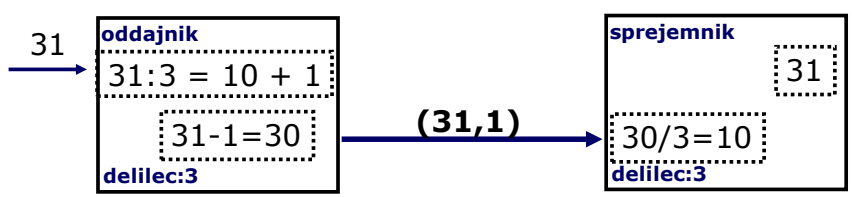
- Eternet okvir:



Uvod v CRC

- Uporablja se pri prenosu in shranjevanju podatkov (zip, ethernet, CD, disk...).
- Osnovna ideja:
  - **Preverjanje deljivosti** zaporedja bitov z zaporedjem *delilnih bitov*.
  - **Oddajnik spremeni** zaporedje bitov tako, da **postane deljivo** (brez ostanka) z nekim **v naprej** določenim zaporedjem.
  - **Sprejemnik** dobi spremenjeno zaporedje in **preveri, če je deljivo** z v naprej določenim zaporedjem.

Intuitivni primer:



Osnove: Aritmetika po modulu 2

- V CRC kodiranju se uporablja aritmetika po modulu 2.
- Operacija **xor**:

A	**0**	**1**	**1**	**0**
B	**0**	**0**	**1**	**1**
A xor B	**0**	**1**	**0**	**1**

- **Odštevanje** je **enako seštevanju**: ***A-B = A+B***
- **Deljenje** po modulu 2 je podobno deljenju pri desetiških številih, namesto odštevanja uporabimo operacijo *xor*.

- Primeri:http://www.pccl.demon.co.uk/papers/modulo2.html

Naloga1: Primer deljenja po modulu 2

Delite besedo 100100111 z besedo 10011 in izračunajte ostanek pri deljenju.

Uvod v CRC

- Ciklični kod imenujemo tudi **polinomski kod**.
- **Zaporedje *r* binarnih simbolov** obravnavamo kot **polinom** stopnje *(r-1)*.
- Primer:
  - k+1 simbolov:

    $$b_k, b_{k-1}, b_{k-2}, \ldots, b_1, b_0$$

  - Polinom k-te stopnje:

    $$P_k(x) = b_k x^k + b_{k-1} x^{k-1} + \ldots, b_1 x^1 + b_0 x^0$$

  - biti:

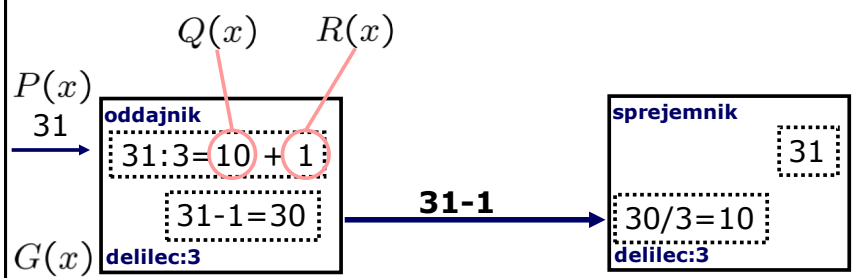
    $$b_i \in \{0, 1\}$$

Naloga2: Polinomski zapis

Zaporedju šestih simbolov 110110 priredi polinom primerne stopnje, dobljeni polinom množite z x^2 in ga zapišite spet binarno.

Spomnimo se primera z decimalnimi števili

Intuitivni primer:



Princip CRC: Oddajnikova stran

Predprocesiranje:

1. Vhodno besedo dolžine (k+1) bitov predstavimo s $P_k(x)$.
2. Delilno besedo dolžine (r+1) bitov predstavimo z $G_r(x)$.

Kodiranje:

1. Vhodni besedi dodamo (r) bitov z vrednostjo 0: $P_k(x) \cdot x^r$
2. Izračunamo ostanek deljenja $R(x)$ pri $\frac{P_k(x)x^r}{G_r(x)}$:
   $\frac{P_k(x)\cdot x^r}{G_r(x)} = Q(x) + \frac{R(x)}{G_r(x)}$
3. Odštejemo ostanek od razširjenega polinoma in oddamo v komunikacijski kanal:
   $T(x) = P_k(x)x^r - R(x) = P_k(x)x^r + R(x)$

Naloga3:

S polinomskim zapisom pokažite, da je ciklično kodirano sporočilo deljivo z generatorskim polinomom brez ostanka.

CRC: kako v praksi izvedemo kodiranje?

- Vhodno besedo 110111 dolžine (k+1) bitov (k=5): $P_k(x)$
- Delilno besedo 11001 dolžine (r+1) bitov (r=4): $G_r(x)$

1. Vhodni besedi dodamo (r) bitov z vrednostjo 0: $P'_k(x) = P_k(x)x^r$
   - 110111 0000
2. Izračunamo ostanek deljenja $R(x)$ pri $\frac{P_k(x)x^r}{G_r(x)}$
   - 110111 0000 / 11001 = ?
3. Dodane bite v razširjeni besedi zamenjamo z ostankom $R(x)$.
   - odštevanje je enako seštevanju (xOR)

$$T(x) = P_k(x)x^r - R(x) = P_k(x)x^r + R(x)$$

Naloga4: Oddajnikova stran

Vhodno besedo 110111 zakodirajte s cikličnim kodiranjem, pri tem pa uporabite delilno zaporedje 11001.(1) Zapiši vhodno besedo in delilno zaporedje s polinomi. (2) Zapiši zakodirano besedo in njen polinom.

Princip CRC: Sprejemnikova stran

1. Oddajnik je oddal besedo: $T(x) = P_k(x)x^r + R(x)$
2. Med prenosom se lahko zgodi napaka na nekaterih bitih:
   $T'(x) = T(x) + E(x)$
3. Sprejemnik dobi besedo $T'(x)$ in jo deli z $G_r(x)$.
   $\frac{T'(x)}{G_r(x)} = \frac{T(x)+E(x)}{G_r(x)}$
4. Če do napake ni prišlo, $E(x) = 0$, potem je deljenje celoštevilčno.
5. Primer polinoma, ki odkriva enojno, dvojno in izbruh napak na 16 zaporednih simbolih ali manj:

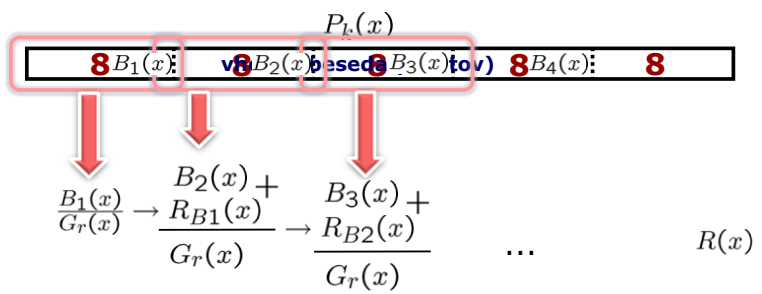
$$P_{CRC-16} = x^{16} + x^{12} + x^5 + 1$$

Naloga5: Sprejemnikova stran

V prejšnji nalogi smo izračunali polinom:
$T(x) = x^9 + x^8 + x^6 + x^5 + x^4 + x^2 + x$
Kolikšen je ostanek pri deljenju $\frac{T(x)}{G(x)}$, če $G(x) = x^4 + x^3 + 1$?

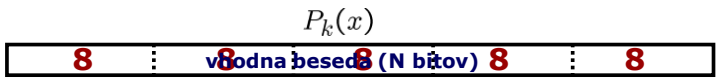
Ciklično preverjanje s tabelo ostankov

$\frac{P_k(x)}{G_r(x)} = Q(x) + \frac{R(x)}{G_r(x)}$ Koliko je ostanek $R(x)$?



Pomembno: *Ostanek lahko računamo s sekvenčnim algoritmom. Sekvenčno deljenje 8 bitov.*

Ciklično preverjanje s tabelo ostankov



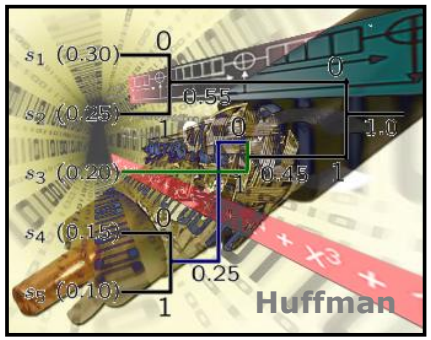
- **Ideja:**
  - Zaporedoma jemljemo **bloke** po **8 bitov**.
  - Prištejemo **del ostanka** iz prejšnjega koraka **k novemu bloku**.
  - Izračunamo **nov ostanek**.
  - Nadaljujemo dokler ne pregledamo vseh blokov.
- **Poenostavitev:**
  - Vemo, da delimo samo 8 bitna števila.
  - V naprej izračunamo (**tabeliramo**) **vse možne ostanke** pri **deljenju 8 bitnih števil** z izbranim generatorskim polinomom.

Naloga6:

Izdelajte postopek cikličnega kodiranja na osnovi tabele ostankov za polinom stopnje 8 ($G_8(x)$).

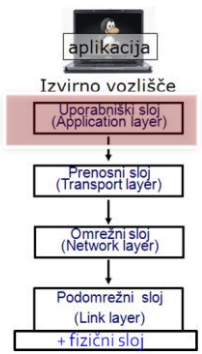
Zgoščevanje podatkov

1. Uvod
2. Huffmanov kod.
3. Odvečnost in koristnost.
4. Naloge se navezujejo na (Kovačič 2001, pp. 193 -- 202)



Uvod

- **Kodiranje**: postopek zgoščevanja podatkov.
- **Kod**: pravilo za postopek zgoščevanja
- Aplikacijski sloj:
  - Deflate (PKZIP)
  - JPEG
  - MP3
  - Huffyuv (video kodek)
- Zakaj kodiramo?
  1. zanesljivost prenosa
  2. zanesljivost shranjevanja
  3. izkoristek pomnilnika (zgoščevanje)



Primer: Zgoščevanje besedila

i	a_i	p_i
1	a	0.0575
2	b	0.0128
3	c	0.0263
4	d	0.0285
5	e	0.0913
6	f	0.0173
7	g	0.0133
8	h	0.0313
9	i	0.0599
10	j	0.0006
11	k	0.0084
12	l	0.0335
13	m	0.0235
14	n	0.0596
15	o	0.0689
16	p	0.0192
17	q	0.0008
18	r	0.0508
19	s	0.0567
20	t	0.0706
21	u	0.0334
22	v	0.0069
23	w	0.0119
24	x	0.0073
25	y	0.0164
26	z	0.0007
27	–	0.1928

i	a_i	p_i	$h(p_i)$
1	a	.0575	4.1
2	b	.0128	6.3
3	c	.0263	5.2
4	d	.0285	5.1
5	e	.0913	3.5
6	f	.0173	5.9
7	g	.0133	6.2
8	h	.0313	5.0
9	i	.0599	4.1
10	j	.0006	10.7
11	k	.0084	6.9
12	l	.0335	4.9
13	m	.0235	5.4
14	n	.0596	4.1
15	o	.0689	3.9
16	p	.0192	5.7
17	q	.0008	10.3
18	r	.0508	4.3
19	s	.0567	4.1
20	t	.0706	3.8
21	u	.0334	4.9
22	v	.0069	7.2
23	w	.0119	6.4
24	x	.0073	7.1
25	y	.0164	5.9
26	z	.0007	10.4
27	–	.1928	2.4
		$\sum_i p_i \log_2 \frac{1}{p_i}$	4.1

Koliko bitov potrebujemo za en znak?

4 bit ... 2^4=16 znakov
5 bit ... 2^5=32 znakov

Entropija izmeri informacijsko vsebino:
koliko bitov rabimo za kodiranje glede na verjetnost s katero se pojavi nek znak.

Če bi bila enakomerna porazdelitev:

$$H(X) = -\sum_{i=1}^{27} \frac{1}{27} log_2 \frac{1}{27} = 4.75 \text{ bit/znak}$$

Ker ni enakomerna:

$$H(X) = -\sum_{i=1}^{27} p_i log_2 p_i = 4.1 \text{ bit/znak}$$

Matematična formulacija

- Imamo znake, ki so zapisani s **simboli** s_i .
- Poznamo množico simbolov S z N simboli:

$$S = \begin{pmatrix} s_1 & s_2 & \dots & s_i & \dots & s_N \\ p_1 & p_2 & \dots & p_i & \dots & p_N \end{pmatrix} \quad , \; p_i = p(s_i)$$

- Vsakemu simbolu priredimo binarno **kodno besedo** w_i .
  npr: $w_1 = 01, w_2 = 1010, itd.$
- Kodne besede so različno dolge, npr., beseda $w_2 = 1010$ je dolga $n_2 = 4$ binarnih simbolov.
- Množici **kodnih besed** pravimo **kod** W:
  $W = \{w_i\}, \; i = 1, 2, \dots, N$

Matematična formulacija kodiranja

- **Poznamo** množico simbolov S z N simboli:

$$S = \begin{pmatrix} s_1 & s_2 & s_i & s_N \\ p_1 & p_2 & p_i & p_N \\ w_1 & w_2 & w_i & w_N \\ n_1 & n_2 & n_i & n_N \end{pmatrix}$$

simboli

kodne besede

dolž. kodne bes.

- **Povprečna dolžina:** $\bar{n} = \sum_{i=1}^{N} p_i n_i$

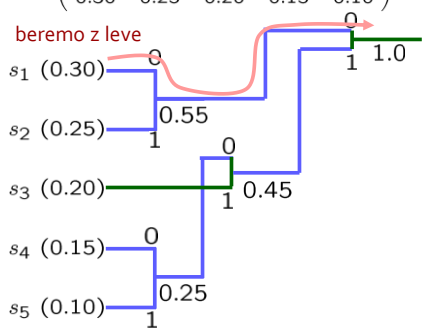
- **Problem:** Kako **določiti** take **kodne besede** w_i , ki **minimizirajo povprečno dolžino** prenešenih besed, in hkrati zadostijo pogojem:
  1. regularnosti – različnim s_i pripadajo različni w_i .
  2. enoznačnosti – sporočilo prebrati možno le na en način.
  3. trenutnosti – vsako kodno besedo se da dekodirati takoj po prejetju.

HUFFMANOVO KODIRANJE!

Postopek Huffmanovega kodiranja:

1. Uredi simbole po padajočih verjetnostih.
2. Združuj na vsakem koraku po dva simbola z najmanjšo verjetnostjo in seštej pripadajoči verjetnosti.
3. Preuredi listo simbolov po padajočih verjetnostih in se vrni na korak 1.

$$S = \begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ 0.30 & 0.25 & 0.20 & 0.15 & 0.10 \end{pmatrix}$$



pišemo z desne

$s_1, w_1 = 00$

$s_2, w_2 = 01$

$s_3, w_3 = 11$

$s_4, w_4 = 100$

$s_5, w_5 = 101$

Odvečnost in koristnost koda

- **Koristnost E:** razmerje med informacijsko vsebino signala in povprečnim številom bitov s katerimi kodiramo signal.

$$E = \frac{H(S)}{\bar{n}}$$

entropija: $H(S) = -\sum_{i=1}^{N} p_i log_2(p_i)$

povprečna dolžina: $\bar{n} = \sum_{i=1}^{N} p_i n_i$

$$S = \begin{pmatrix} s_1 & s_2 & & s_i & & s_N \\ p_1 & , \; p_2 & , \ldots , & p_i & , \ldots , & p_N \\ n_1 & n_2 & & n_i & & n_N \end{pmatrix}$$

- **Odvečnost** (redundanca): $R = 1 - E$

Naloga1: Huffmanovo kodiranje

Izračunajte Huffmanov kod za naslednjo množico simbolov in izračunajte odvečnost koda *E*. Zakodirajte sporočilo: abeceda

$$S = \begin{pmatrix} a & b & c & d & e & f \\ 0.23 & , \; 0.22 & , \; 0.3 & , \; 0.16 & , \; 0.05 & , \; 0.04 \end{pmatrix}$$

$$E = \frac{H(S)}{\bar{n}}$$
$$R = 1 - E$$

Naloga2: Huffmanovo kodiranje

Izračunajte Huffmanov kod, ki optimalno kodira naslednje binarno sporočilo, tako, da skupaj vzamete po tri zaporedne bite. Zakodirajte sporočilo in izračunajte odvečnost koda *E*.

111 101 101 101 010 000 000

$$E = \frac{H(S)}{\bar{n}}$$
$$R = 1 - E$$

Naloga3: Huffmanovo kodiranje

Sestavite Huffmanov kod za binarna sporočila, če pride v povprečju v sporočilih na eno enico devet ničel in če kodiramo po dva zaporedna bita. Izračunaj redundanco.
Zakodiraj sporočilo 000001 – koliko bitov prihranimo?

$$E = \frac{H(S)}{\bar{n}}$$
$$R = 1 - E$$

Konec.

- Hvala.