Aplikovaná informatika

Podklady předmětu
Aplikovaná informatika
pro akademický rok 2013/2014
Radim Farana

3

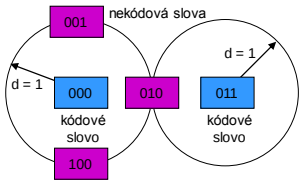
Obsah

- Detekce chyb, Hammingova vzdálenost.
- Kontrolní a samoopravné kódy.
  - Lineární kódy.
  - Hammingovy kódy.
  - Opakovací kódy.
  - Cyklické kódy.

Detekce chyb

- Množinu všech slov rozdělíme na slova kódová a slova nekódová.
- *t*-násobná chyba změní kódové slovo na nekódové, pokud se dvě kódová slova liší ve více než *t* znacích.
- Hammingova vzdálenost je počet znaků ve kterých se dvě kódová slova liší.
- Hammingova vzdálenost kódu *d* je nejmenší z nich.

Hammingova vzdálenost



Hamming, Richard Wesley

* 11. 2. 1915 Chicago, Il. USA
+ 7. 1. 1998 Monterey, Cal. USA
http://cm.bell-labs.com/cm/cs/alumni/hamming/

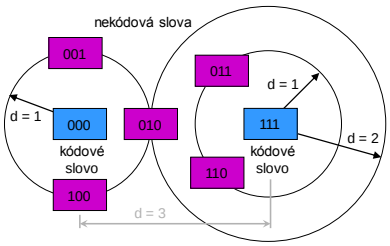
Kód odhaluje *t*-násobné chyby, pokud je Hammingova vzdálenost kódu $d > t$.

Označení kódů (blokových): (n, k)-kód

počet znaků / počet informačních znaků

$(4, 3)$-kód má jeden kontrolní znak, je schopen mít $d = 2$.

Opravování chyb



Kód opravuje *t*-násobné chyby, pokud je Hammingova vzdálenost kódu $d > 2.t$.

Kód dvourozměrné kontroly parity

- Informační znaky zapíšeme do matice typu (p, q).
- Každému řádku přidáme jeden symbol kontroly parity řádku, podobně každému sloupci kontrolu parity sloupce.
- Paritě sloupce parit řádků pak znak „*kontrola kontrol*“, volený tak, aby i parita výsledné matice byla sudá.
- Např. pro $p = 7$ a $q = 3$ obdržíme ASCII (32, 21)-kód. Tento kód opravuje jednoduché chyby.

Příklad

101	0
000	0
001	1
010	1
111	1
111	1
000	0
110	0

kontrola parity řádků

kontrola kontrol

kontrola parity sloupce

Lineární kódy (maticové kódy)

- Kódové slovo chápeme jako řádkový vektor **v** = [0 0 1].
- Lineární kombinací libovolného počtu kódových slov vznikne opět kódové slovo.
- Kód je možno popsat pomocí generující matice (kterou tvoří báze kódu).

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{v} = \mathbf{z}.\mathbf{G}$$

Systematické kódy

- Informační slovo tvoří začátek kódového slova. $\mathbf{G} = [\,\mathbf{E}\,|\,\mathbf{B}\,]$
- Určení informačního slova (dekódování) je triviální.
- Je možno snadno určit kontrolní matici kódu $\mathbf{H} = [\,-\mathbf{B}^T|\,\mathbf{E}'\,]$
- A syndrom přijatého slova $\mathbf{s} = \mathbf{H}.\mathbf{v}^T$
- Nenulový syndrom indikuje chybu.

Hammingovy kódy

- Opravují jednoduché chyby a
- jsou perfektní = při daných vlastnostech mají minimální možnou redundanci.
- Kód s m kontrolními znaky (m = 2, 3, …) má délku $n = 2^m - 1$.
- Sloupce kontrolní matice tvoří binární rozvoj čísel 1, 2, …, 2^m - 1
- Nenulový syndrom je binárním rozvojem pozice chyby.

Cyklické kódy (polynomické kódy)

- Jsou podtřídou lineárních kódů. Kódové slovo chápeme jako zápis polynomu.
- Cyklickým posunem znaků kódového slova vznikne opět kódové slovo.

$a_0 + a_1 z + a_2 z^2 + \ldots a_{n-1} z^{n-1} \sim a_0 a_1 a_2 a_3 \ldots a_{n-1}$

- Kromě generující matice mohou být popsány také generujícím polynomem.
- Jsou schopny dobře detekovat (opravovat) i shlukové chyby.

Cyklické kódy

- Cyklický posun odpovídá násobení proměnnou z.
- Přesun koeficientu u nejvyšší mocniny na začátek kódového slova vyřešíme zavedením: $z^n = 1, z^{n+1} = z, z^{n+2} = z^2, \ldots$
- Dělení polynomu a(z) polynomem b(z) chápeme jako určení podílu q(z) a zbytku r(z). $a(z) = q(z) \cdot b(z) + r(z)$ a $\deg\ r(z) < \deg\ b(z)$

Okruh polynomů modulo q(z)

- Významný je pojem ***okruh polynomů modulo*** q(z): $T / q(z)$
  - Kde je T - těleso vzniklé z abecedy T, u binárních kódů pracujeme s tělesem $Z_2 = \{0, 1\}$,
  - q(z) – polynom proměnné z nad tělesem T, koeficienty polynomu jsou prvky tělesa T.
- Základní operace v okruhu polynomů modulo q(z) jsou pak:
  - Sčítání a(z) + b(z) – stejné jako sčítání polynomů.
  - Násobení a(z) ⊗ b(z) je však definováno jako zbytek po dělení polynomu a(z).b(z) polynomem q(z).

Generující polynom g(z)

- Pokud zvolíme $q(z) = z^n - 1$ vznikne okruh polynomů, ve kterém platí $z^n = 1$.
- Polynomy patřící do tohoto okruhu pak definují jednotlivá kódová slova cyklického kódu.
- Generující polynom cyklického (n, k)-kódu je polynom stupně $n - k$ v tomto kódu, který je dělitelem polynomu $(z^n - 1)$

Generující matice cyklického kódu

- Generující matice cyklického kódu vznikne cyklickým posunem koeficientů generujícího polynomu:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k-1} & g_{n-k} & 0 & \dots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & \dots & g_{n-k} \end{bmatrix}$$

(k-1 nul v každém řádku; k-1 řádků)

Její řádky tvoří polynomy:

$$g(z)$$
$$z \cdot g(z)$$
$$\vdots$$
$$z^{k-1} \cdot g(z)$$

Kontrolní polynom h(z)

$$h(z) = (z^n - 1) : g(z)$$

- Kontrolní polynom:
- Kontrolní matici cyklického (n, k)-kódu získáme cyklickými posuvy koeficientů kontrolního polynomu čteného od nejvyšší mocniny:

$$H = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 & h_k & h_{k-1} & \dots & h_1 & h_0 \\ 0 & 0 & \dots & 0 & h_k & h_{k-1} & \dots & h_1 & h_0 & 0 \\ \vdots & & & & & & & & & \\ h_k & h_{k-1} & \dots & & & h_1 & h_0 & \dots & & 0 \end{bmatrix}$$

($n-k+1$ sloupců)

Pro každý polynom $v(z)$, pro který platí:

$$v(z) \cdot h(z) = 0$$

splňuje kontrolní matice podmínku:

$$\mathbf{H} \cdot \mathbf{v}^T = \mathbf{0}^T$$

Realizace cyklických kódů

- systematické kódování:
  - kódové slovo čteme pozpátku (neboli polynomy zapisujeme od nejvyšší mocniny, tedy opačně než jsme je zapisovali dosud).
  - Z informačních bitů $u_0\, u_1 \ldots u_{k-1}$ vytvoříme polynom: $u(z) = u_0 z^{n-1} + u_1 z^{n-2} + \ldots + u_{k-1} z^{n-k}$
  - Tento polynom dělíme generujícím polynomem $g(z)$: $u(z) = q(z)g(z) + r(z)$ kde je deg $r(z) < n-k$
  - Odečtením zbytku vznikne kódové slovo: $q(z)g(z) = u(z) - r(z)$

Realizace cyklických kódů

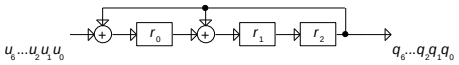
- Protože v binární aritmetice platí , polynom $u(z)$ obsahuje jen koeficienty u mocniny $n - k$ nebo vyšší, zatímco zbytek koeficienty nižší, pak při označení $r(z) = r_k z^{n-k-1} + r_{k+1} z^{n-k-2} + \ldots + r_{n-1} z + r_n$
- vyšleme kódové slovo $u_0 u_1 \ldots u_{k-1} r_k r_{k+1} \ldots r_n$
- Kódové slovo je dělitelné $g(z)$ beze zbytku.
- Pod označením CRC-kódy (Cyclic Redundance Code) mají široké použití.

Typické CRC kódy

počet kontrolních bitů	označení	generující polynom	použití
8	LRCC–8	z^8+1	kontrolní Byte je součet datových Byte modulo 2
12	CRC–12	$z^{12}+z^{11}+z^3+z^2+z+1$	používá se pro šestibitové znaky
16	LCRC–16	$z^{16}+1$	kontrolní součet dvojic Byte (Word) modulo 2
16	CRC–16	$z^{16}+z^{15}+z^2+1$	binární synchronní protokol
16	CRC–16 reverzní	$z^{16}+z^{14}+z+1$	
16	SDLC	$z^{16}+z^{12}+z^5+1$	linkový protokol IBM, standard CCITT
16	SDLC reverzní	$z^{16}+z^{11}+z^4+1$	
32	CRC–32	$z^{32}+z^{26}+z^{23}+z^{22}+$ $+z^{16}+z^{12}+z^{11}+z^{10}+$ $+z^8+z^7+z^5+z^4+$ $+z^2+z+1$	Ethernet, HDLC, ZMODEM

Hardwarová realizace cyklických kódů

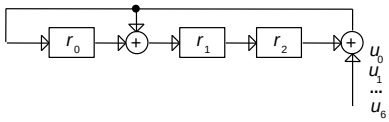
- pro Hammingův (7, 4)-kód s generujícím polynomem $q(x) = x^3 + x + 1$ je dělení generujícím polynomem snadno realizovatelné pomocí dvou binárních sčítaček a tří posuvných registrů



- Do obvodu vstupují koeficienty polynomu $u(z) = u_0 z^6 + u_1 z^5 + ... + u_6$ a vystupují koeficienty podílu $q(z) = q_0 z^6 + q_1 z^5 + ... + q_6$.
  Po vystoupení posledního koeficientu zůstávají v registrech koeficienty zbytku $r(z) = r_2 z^2 + r_1 z + r_0$.

Hardwarová realizace cyklických kódů

- Podíl q(z) je pro nás nepodstatný, zajímá nás pouze zbytek r(z), přesunem vstupu informačních bitů na konec obvodu, získáme zbytek r(z) ihned po průchodu informačních bitů obvodem



Příklad

- Hammingův (7, 4)-kód není cyklický. Např. cyklickým posunem slova 1101001 (první řádek generující matice) dostaneme nekódové slovo 1110100 (jeho syndrom je 101). Protože sloupce kontrolní matice Hammingova kódu můžeme psát v libovolném pořadí, uspořádáme je tak, aby vznikl cyklický kód.

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

Příklad

- Sloupce tvoří všechna slova délky 3 kromě 000. Místo slov budeme psát polynomy stupně nejvýše 2 v pomocné proměnné x (její označení je voleno z důvodu odlišení od proměnné z). Chceme najít pořadí, v jakém zapsat všechny nenulové polynomy $a + bx + cx^2$ jako sloupce $\begin{bmatrix} c \\ b \\ a \end{bmatrix}$

Příklad

- matice **H** tak, aby vznikl cyklický kód. $Z_2 / q(x)$
  K tomu použijeme okruh
  kde je q(x) polynom třetího stupně, takže prvky okruhu jsou právě naše polynomy.
  Jako vhodná volba se ukazuje:

$$q(x) = x^3 + x + 1$$

neboť platí $x^3 + x + 1 = 0$

Příklad

- takže mocninami x^i vyjádříme naše polynomy

$$x^0 = 1$$
$$x^1 = x$$
$$x^2 = x^2$$
$$x^3 = x + 1$$
$$x^4 = x^2 + x$$
$$x^5 = x^2 + x + 1$$
$$x^6 = x^2 + 1$$

Příklad

- Kontrolní matici **H** nyní uspořádáme podle těchto mocnin:

$$\mathbf{H} = \begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Příklad

- Kód s touto kontrolní maticí sestává ze všech polynomů:
  $v(z) = v_0 + v_1 z + \ldots + v_6 z^6$
  pro které platí:

$$\mathbf{H} \begin{bmatrix} v_0 \\ v_1 \\ \cdot \\ \cdot \\ \cdot \\ v_6 \end{bmatrix} = \begin{bmatrix} 1 & x & x^2 & \cdot & \cdot & \cdot & x^6 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ \cdot \\ \cdot \\ \cdot \\ v_6 \end{bmatrix} = v_0 + v_1 x + v_2 x^2 + \ldots + v_6 x^6 = 0$$

Příklad

- neboli $v(x) = 0$ v okruhu $Z_2 / (x^3 + x + 1)$
  a tento kód je cyklický, přitom má kontrolní matice jako sloupce všechna nenulová slova délky 3, takže je to Hammingův (7, 4)-kód.

Příklad

- Generující polynom je stupně 7 – 4 = 3 a je to jediný takový polynom
  $g(z) = z^3 + z + 1$

Generující matice je tedy:

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

Příklad

- Kontrolní polynom h(z) určíme ze vztahu
  $h(z) = (z^n - 1) : g(z)$
  $h(z) = (z^7 - 1) : (z^3 + z + 1) = z^4 + z^2 + z + 1$

Ten určuje kontrolní matici:

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