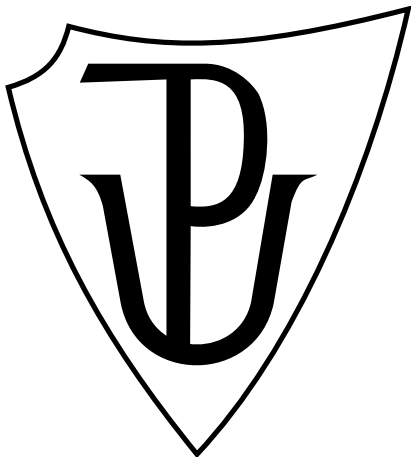
Teorie informace a kódování (KMI/TIK)

Bezpečnostní kódy



Lukáš Havrlant

Univerzita Palackého

13. listopadu 2012

Konzultace

- V pracovně 5.076.
- Každý čtvrtek 9.00–11.00.
- Emaily:
  - lukas@havrlant.cz
  - lukas.havrlant@upol.cz
- Web: `http://tux.inf.upol.cz/~havrlant/`

Doporučená literatura

- Adámek J.: Foundations of Coding. J. Wiley, 1991.
- Adámek J.: Kódování. SNTL, Praha, 1989. (lze stáhnout na `http://notorola.sh.cvut.cz/~bruxy/`)
- Ash R. B.: Information Theory. Dover, New York, 1990.

Úvod

- Při přenosu dat může dojít k chybě $\longrightarrow$ jak odhalit chybu? (znáte: CRC, hash) Jak opravit chybu?
- Chyba může být dvojího typu:
  1. Záměna znaku: 101010 $\longrightarrow$ 101000 (o tento typ se budeme zajímat)
  2. Vytvoření nového znaku: 101010 $\longrightarrow$ 1010101. Pohlcení znaku: 101010 $\longrightarrow$ 10110
- Základní princip: redundance informace.
- Čeština je hodně redundantí: „opakkvání“ $\longrightarrow$ „opakování“, ale „sekers“ $\longrightarrow$ „sekera“ nebo „sekery“?
- Všechna rodná čísla vytvořená od roku 1986 včetně jsou dělitelná jedenácti.

Příklad kódování objevující jednu chybu

- Kontrola sudé parity: kód doplní na konec binárního slova $0/1$ tak, aby slovo mělo sudý počet jedniček.
- Vstupem je slovo $w \in \{0,1\}^+$, doplníme $c \in \{0,1\}$, vznikne slovo wc.
- Pokud $w = 001$, pak $c = 1$ a $wc = 0011$.
- Pokud $w = 101$, pak $c = 0$ a $wc = 1010$.
- Kód objevuje 1 chybu: pokud přijmeme 1011, nastala chyba.
- Kód nerozpozná dvě chyby! Vyšleme 1010, přijmeme 1111.

Příklad kódování opravující jednoduché chyby

- Opakující kód – symbol odešleme třikrát.
- $0 \longrightarrow 000$; $1 \longrightarrow 111$.
- Pokud přijmeme 010, poslali jsme původně 000 (nebo nastala více než jedna chyba).

Cíl

- Cíl: sestrojit kód, který objeví/opraví co nejvíce chyb při co nejnižší redundanci.
- Uvažujeme pouze blokové kódy.

Kódová a nekódová slova

- Zdrojová abeceda A, kódová abeceda B, délka kódového slova je n, pak pro kód C platí: $C \subseteq B^n$.
- $B^n = \{b_1 b_2 \dots b_n | b_i \in B \text{ pro } i = 1, 2 \dots n\}$
- C je množina kódových slov, $B^n \setminus C$ množina nekódových slov.
- Přijmeme-li nekódové slovo, tak během přenosu nastala chyba.
- Pokud přijmeme kódové slovo, tak během přenosu. . . ?
- Kontrola sudé parity: pro $B = \{0, 1\}, n = 4$ máme
  - $B^4 = \{0000, 0001, 0010, 0011, 0100, 0101, \dots\}; \quad |B^4| = 2^4 = 16$
  - $C = \{0000, 0011, 0101, 0110, \dots\}; \quad |C| = 8$

Informační poměr kódu (rate)

- Máme slovo délky k. Přidáme symboly z kódové abecedy a získáme slovo o délce n.
- Původním k symbolům říkáme *informační symboly*.
- $n - k$ symbolům říkáme *kontrolní symboly*.
- Poměr $R(C) = \frac{k}{n}$ měří redundanci kódu.

Definice: Nechť $C \subseteq B^n$ je blokový kód o délce n. Jestliže existuje bijektivní zobrazení $\varphi : B^k \to C$, řekneme, že kód C má k informačních symbolů a $n - k$ kontrolních. *Informační poměr kódu* $R(C)$ je

$$R(C) = \frac{k}{n}.$$

Příklady informačních poměrů

- **Opakovací kód**
  - Kód C_n má na abecedě B kódová slova $\{a^n \mid a \in B\}$
  - Pro $B = \{0, 1\}$ máme $C_3 = \{000, 111\}$, $C_5 = \{00000, 11111\}$
  - $k = 1; \quad R(C_n) = \frac{1}{n}$
- **Kontrola sudé parity**
  - Pro kód délky n platí: $C_n = \{w \text{ má sudý počet } 1 \mid w \in \{0, 1\}^n\}$
  - $C_3 = \{000, 011, 101, 110\}$
  - Pro C_3: $|C| = 4, k = 2; \quad R(C_3) = \frac{2}{3}$
  - Pro C_n: $|C| = 2^{n-1}, k = n - 1; \quad R(C_n) = \frac{n-1}{n}$

Hammingova vzdálenost

Definice: Pro slova $u = u_1 \cdots u_n, v = v_1 \cdots v_n \in B^n$ definujeme zobrazení

$$\delta(u, v) = |\{i \,|\, 1 \leq i \leq n, u_i \neq v_i\}|.$$

Tj. $\delta(u, v)$ vrací počet pozic, na kterých se slova u a v liší. Zobrazení δ nazýváme **Hammingova vzdálenost**. δ je metrikou na B^n:

$$\begin{aligned} \delta(u, v) &\geq 0 \quad (= 0 \text{ iff } u = v), \\ \delta(u, v) &= \delta(v, u), \\ \delta(u, v) &\leq \delta(u, w) + \delta(w, v). \end{aligned}$$

$\delta(011, 101) = 2, \quad \delta(011, 111) = 1, \quad \delta(0110, 1001) = 4, \; \delta(0123, 0145) = 2, \quad \delta(01, 01) = 0$

Důkaz trojúhelníkové nerovnosti

Chceme dokázat: $\delta(u, v) \leq \delta(u, w) + \delta(w, v)$. Označme:

$$\begin{aligned} U &= \{i \,|\, 1 \leq i \leq n, u_i \neq w_i\}, \\ V &= \{i \,|\, 1 \leq i \leq n, w_i \neq v_i\}. \end{aligned}$$

Pak $U \cup V$ je množina indexů, na kterých se u může lišit od v. Pro všechna $i \in \{1, 2, \ldots, n\} \setminus (U \cup V)$ platí, že $u_i = w_i = v_i$. Dále

$$\delta(u, v) \leq |U \cup V|.$$

$\delta(u, v)$ může být menší, pokud $u_i \neq w_i \wedge w_i \neq v_i$, ale $u_i = v_i$. Pak $i \in U \cup V$, ale $i \notin \{j \,|\, 1 \leq j \leq n, u_j \neq v_j\}$. Dostáváme:

$$\delta(u, v) \leq |U \cup V| \leq |U| + |V| = \delta(u, w) + \delta(w, v). \quad \square$$

Minimální vzdálenost

Definice: Minimální vzdálenost kódu C, zapíšeme $d(C)$, je minimální vzdálenost dvou různých slov z C:

$$d(C) = \min\{\delta(u, v) \mid u, v \in C, u \neq v\}.$$

Příklady:

$$\begin{aligned} C_1 &= \{0011, 1010, 1111\}, & d(C_1) &= 2 \\ C_2 &= \{0011, 1010, 0101, 1111\}, & d(C_2) &= 2 \\ C_3 &= \{000, 111, 100\}, & d(C_3) &= 1 \end{aligned}$$

Koule

Definice: Nechť C je nějaký kód o délce n a $u \in C$. Pak $B_q(u) = \{v \in B^n \mid \delta(u, v) \leq q\}$ nazveme koule se středem v u a poloměrem q.

Důsledek: Kód C má minimální vzdálenost $d(C) \geq d$ právě tehdy, když pro všechna $u \in C$ platí

$$B_{d-1}(u) \cap C = \{u\}.$$

Objevování chyb

- t chyb nastane, pokud pošleme slovo u, přijmeme slovo v a $\delta(u, v) = t$. (Slovo u se od v liší v t symbolech.)

  Definice: Kód C *objevuje* d chyb, pokud při odeslání kódového slova a vzniku $t \leq d$ chyb, přijmeme vždy nekódové slovo.

  Věta: Kód C *objevuje* d chyb právě když $d(C) > d$.

Největší d takové, že C objevuje d chyb je $d(C) - 1$.

Příklady

- **Kontrola sudé parity:**
  - $d(C) = 2$, takže kód objevuje 1 chybu.
  - Proč $d(C) = 2$? Nechť $u, v \in B^{n-1}$, $u \neq v$. Slova zakódujeme, dostaneme ua, vb, kde $a, b \in B$.
    - Pokud $\delta(u, v) \geq 2$, pak jistě $\delta(ua, vb) \geq 2$.
    - Pokud $\delta(u, v) = 1$, pak právě jedno ze slov obsahuje lichý počet jedniček a tedy $a \neq b$ a $\delta(ua, vb) = 2$.

- **Opakovací kód:**
  - $d(C_n) = n$, takže kód objevuje $n - 1$ chyb.

Oprava chyb

- Pokud přijmeme nekódové slovo v, *opravíme* ho na takové kódové slovo u, které je slovu v nejblíže.

- $\longrightarrow$ pro každé nekódové slovo musí existovat jediné kódové slovo, které je mu nejblíže.

  Definice: Kód C opravuje d chyb pokud pro každé kódové slovo $u \in C$ a každé slovo $v \in B^n$ takové, že $\delta(u, v) \leq d$, platí, že slovo u má od slova v nejmenší vzdálenost ze všech slov z C.
  Tedy pokud $u \in C$ a $\delta(u, v) \leq d$, pak $\delta(u, v) < \delta(w, v)$ pro všechna $w \in C, w \neq u$.

Oprava chyb podruhé

Věta: C opravuje d chyb právě tehdy, když $d(C) > 2d$.

- Největší d takové, že C opravuje d chyb je $d = \lfloor (d(C) - 1)/2 \rfloor$. Příklady:
- Kontrola sudé parity má $d(C) = 2$, takže opravuje 0 chyb.
- Opakovací kód C_3 opravuje $(3 - 1)/2 = 1$ chybu.

Důkaz

Věta: C opravuje d chyb právě tehdy, když $d(C) > 2d$.

„$\Leftarrow$“: Nechť $d(C) > 2d$. Dále nechť $u \in C, v \in B^n : \delta(u, v) \leq d$. Sporem předpokládejme, že pro $w \in C, w \neq u$ platí $\delta(w, v) \leq \delta(u, v)$. Použijeme trojúhelníkovou nerovnost:

$$\delta(w, v) + \delta(u, v) \geq \delta(w, u)$$

Přitom ale $\delta(u, v) \leq d$ a zároveň $\delta(w, v) \leq d$, takže:

$$2d = d + d \geq \delta(w, v) + \delta(u, v) \geq \delta(w, u).$$

Dostáváme $2d \geq \delta(w, u)$, což je spor s předpokladem, protože $u, w \in C$.

Důkaz: pokračování

Věta: C opravuje d chyb právě tehdy, když $d(C) > 2d$.

„$\Rightarrow$“: Nechť C opravuje d chyb. Opět sporem předpokládejme, že $d(C) \leq 2d$ a že tak existují $u, v \in C; u \neq v$ taková, že $t = \delta(u, v) \leq 2d$. Rozdělme důkaz na dva případy.
Když je t sudé. Dále nechť w je slovo, které má vzdálenost $t/2$ od obou slov u, v. Slovo w sestrojíme takto: nechť

$$I = \{i \mid u_i \neq v_i\} = \{i_1, i_2, \ldots, i_t\},$$

takže $|I| = t$. Pak $w_i = v_i$ pro všechna $i \in \{i_1, i_2, \ldots, i_{t/2}\}$ a $w_i = u_i$ pro všechna ostatní i. Potom $\delta(u, w) = t/2 = \delta(v, w)$, tedy $\delta(u, w) = \delta(v, w) \leq d$, což je spor s tím, že C opravuje d chyb.

Důkaz: finále

Když je t liché. Pak $t+1$ je sudé, přitom $t+1 \leq 2d$. Stejně jako v předchozím kroku sestavíme slovo w tak, že $w_i = v_i$ pro všechna $i \in \{i_1, i_2, \ldots, i_{(t+1)/2}\}$ a $w_i = u_i$ pro všechna ostatní i. Tzn., že

$$\begin{aligned} \delta(u,w) &= (t+1)/2 \\ \delta(v,w) &= (t-1)/2 \end{aligned}$$

To je ale průšvih, protože $\delta(u,w) > \delta(v,w)$, přitom $\delta(u,w) = (t+1)/2 \leq d$. Jinými slovy: vyslali jsme slovo u, cestou se poškodilo $(t+1)/2 \leq d$ symbolů a my přijali slovo w, které má ale blíže ke slovu v! Kód tak nemůže opravovat d chyb. □

Příklady

- **Kontrola sudé parity:** $d(C) = 2$, kód neopravuje žádnou chybu.
- **Opakovací kód:** $d(C_n) = n$, kód opravuje $\lfloor (n-1)/2 \rfloor$ chyb.

Dvoudimenzionální kontrola sudé parity

Informační symboly umístíme do matice $p-1 \times q-1$, ke které přidáme jeden sloupec a jeden řádek pro kontrolní součty, tj. aby celý řádek/sloupec obsahoval sudý počet jedniček. Poslední bit nastavíme tak, aby celé slovo mělo sudou paritu. Příklad pro $(p, q) = (4, 5)$:

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

Kód opravuje 1 chybu.

Kód dva-z-pěti

- Binární blokový kód s délkou $n = 5$.
- Kódová slova obsahují 2 jedničky. Celkem: $\binom{5}{2} = 10$ možností.
- $00011, 00101, 00110, 01001, 01010, \ldots$
- Populární na reprezentování číslic.
- Dekódování: Můžeme přidat váhu jednotlivým pozicím: 0, 1, 2, 3, 6. Potom můžeme kódové slovo 00101 dekódovat jako: $0 \cdot 0 + 0 \cdot 1 + 1 \cdot 2 + 0 \cdot 3 + 1 \cdot 6 = 8$.
- Trojka je reprezentována dvěma slovy: 10010 a 01100. Druhá možnost se používá pro nulu.

Systematický kód

Definice: Blokový kód C o délce n nazýváme **systematický** kód, pokud pro nějaké $k < n$, pro všechna slova $u \in B^k$, existuje právě jedno slovo $v \in B^{n-k}$ takové, že $uv \in C$ je kódové slovo.

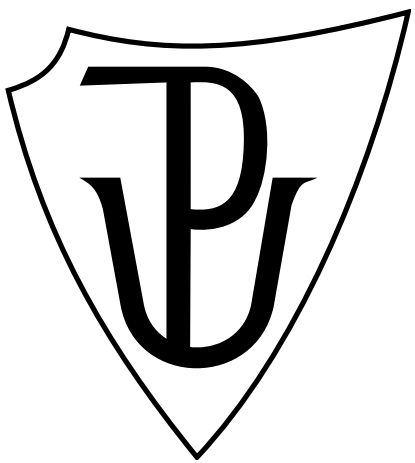
- Opakovací kód a kód sudé parity jsou systematické kódy.
- $R(C) = \frac{k}{n}$

Cvičení

- Jaká je $d(C)$ kódu dva-z-pěti? Jaký je informační poměr?
- Jaká je $d(C)$ dvoudimenzionální kontroly sudé parity? Jaký je informační poměr?
- Za použití dvoudimenzionálního $(4, 5)$-kódu (4 řádky, 5 sloupců) jsme přijali slovo 01001 11101 11100 01100. Je to kódové slovo? Pokud ne, lze opravit?
- Trojúhelníková nerovnost říká, že $\delta(u, v) \leq \delta(u, w) + \delta(w, v)$. Existují navzájem různá slova $u, v, w \in B^n$ pro nějaké B a nějaké n tak, že $\delta(u, v) = \delta(u, w) + \delta(w, v)$?

Teorie informace a kódování (KMI/TIK)

Binární lineární kódy



Lukáš Havrlant

Univerzita Palackého

20. listopadu 2012

Lineární kódy

- Základní myšlenka: ke kódové abecedě B přidáme algebraické operace $+$ a $\cdot$ tak, abychom z B dostali těleso a z B^n vektorový prostor.

- Kódy potom můžeme popsat rovnicemi.

Binární lineární kódy na $Z_2 = \{0, 1\}$

$+$	0	1
0	0	1
1	1	0

a

$\cdot$	0	1
0	0	0
1	0	1

- 0 je neutrální vzhledem k $+$, máme tak inverzní prvky vzhledem k $+$: $0 + 0 = 0$, takže $0 = -0$ a $1 + 1 = 0$, takže $1 = -1$.
- Slova můžeme vidět jako vektory: $1101 = \langle 1, 1, 0, 1 \rangle$
- Skalární násobení: $a \cdot \langle u_1, \ldots, u_n \rangle = \langle a \cdot u_1, \ldots, a \cdot u_n \rangle, a \in Z_2$.
- Sčítání: $\langle u_1, \ldots, u_n \rangle + \langle v_1, \ldots, v_n \rangle = \langle u_1 + v_1, \ldots, u_n + v_n \rangle$.

Kódy jako rovnice

- Důležité kódy jsme schopni popsat rovnicemi:
- Pro slovo x délky n u kódu kontroly sudé parity

$$x_1 + x_2 + \cdots + x_n = 0$$

- Řešením této rovnice jsou všechna kódová slova.
- Opakovací kód $C_n = \{a^n \mid a \in Z_2\}$, např. $C_3 = \{000, 111\}$.

$$\begin{aligned} x_1 + x_n &= 0 \\ &\cdots \\ x_{n-1} + x_n &= 0 \end{aligned}$$

Vektorový podprostor

- Opakování: nechť V je vektorový prostor na množině K. Pak $C \subset V$ je vektorový podprostor, pokud:
  - $\forall u, v \in C : \quad u + v \in C$
  - $\forall a \in K, u \in C : \quad a \cdot u \in C$
- Množina všech řešení systému homogenních rovnic tvoří vektorový podprostor.
- Tj. Z_2^n tvoří vektorový prostor, množina řešení C tvoří vektorový podprostor.

Binární lineární blokové kódy

Definice: Binární blokový kód C se nazývá lineární, pokud $\forall u, v \in C$ platí $u + v \in C$.

- V našem případě: $V = \{0,1\}^n, K = \{0,1\}$
- Proč tam nemusí být podmínka $\forall a \in K, u \in C : \quad a \cdot u \in C$?
- Protože $1 \cdot u = u$ a $0 \cdot u = \mathbf{0} \longrightarrow$ potřebujeme, aby C obsahovalo nulový vektor. Ten obsahuje, protože $u + u = \mathbf{0}$.

Lemma: Pro $n \in \mathbb{N}$, množina C všech řešení homogenního systému rovnic na množině Z_2 je vektorový podprostor Z_2^n. Tedy C je lineární blokový kód.

- Důsledek: pokud popíšeme blokový kód homogenním systémem rovnic, je tento kód lineární.

- Je kód dva-z-pěti lineární? (Pro připomenutí: délka 5, obsahuje vždy dvě 1: 00011, 01010, atd.)

Chybové slovo, Hammingova váha

Definice: Pokud pošleme slovo u a přijmeme slovo v, pak slovo e, které má 1 na pozicích, na kterých se u od v liší, se nazývá *chybové slovo*.

Přitom platí: $v = u + e$ a také $e = v - u\,(= v + u)$.

Definice: *Hammingova váha* slova u je rovna počtu symbolů v u, které se liší od 0.

Definice: *Minimální váha* netriviálního kódu C je nejmenší Hammingova váha slova z C kromě nulového slova.

Vztah minimální váhy a minimální vzdálenosti kódu

Věta: Minimální váha netriviálního kódu C je rovna $d(C)$.

Důkaz: Směr min_weight(C) $\geq d(C)$: Nechť c je minimální váha C. Nechť u je slovo s váhou c. Zřejmě $c = \delta(u, 0 \ldots 0) \geq d(C)$.

Směr min_weight(C) $\leq d(C)$: Nechť $d(C) = \delta(u, v)$ pro nějaká $u, v \in C$. Pak jistě také $u + v \in C$. Přitom Hammingova váha $u + v$ je rovna $\delta(u, v)$, protože $u + v$ má 1 na těch pozicích, na kterých se slova u, v liší. Tedy min_weight(C) $\leq d(C)$.

Celkově tak máme $d(C) =$ min_weight(C).

Kontrolní matice kódu

Definice: Kontrolní matice binárního blokového kódu C o délce n je binární matice H taková, že pro všechny $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ máme

$$H \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

právě tehdy, když $x \in C$.

Matice H je kontrolní maticí kódu C, pokud platí:

$$C = \{x \in \{0,1\}^n \mid Hx^T = \mathbf{0}^T\}$$

Příklady kontrolních matic

- Pokud jsou řádky kontrolní matice nezávislé, H je $n-k \times n$ matice. Kód pak má $n-k$ kontrolních symbolů a n celkových.
- Kontrolní matice kódu kontroly sudé parity o délce n je matice $1 \times n$:

$$H = (1 \ldots 1)$$

- Kontrolní matice pro opakovací kód délky 5 je matice 4×5:

$$H = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Věta o kontrolní matici I

Věta: Pokud binární kód C opravuje jednoduché chyby, pak kontrolní matice kódu C má nenulové a navzájem různé sloupce.

Důkaz: Pokud C opravuje jednoduché chyby, pak $d(C) > 2$ a minimální váha je také > 2. Nechť H je kontrolní matice pro C. Nechť vektor $b^i \in \{0,1\}^n$ má 1 právě na pozici i. Nechť vektor $b^{i,j} \in \{0,1\}^n$ má 1 právě na pozicích i a j. Pokud by matice H měla i-tý sloupec nulový, pak $Hb^{i^T} = \mathbf{0}^T$. Ale b^i je přitom slovo s Hammingovou váhou 1, což je spor s tím, že minimální váha kódu C je > 2.

Pokud by H měla stejné sloupce i a j, pak $Hb^{i,j^T} = \mathbf{0}^T$. ($Hb^{i,j^T}$ je součet sloupců i a j). Nicméně $b^{i,j}$ je slovo s váhou 2, což je opět spor s tím, že minimální vzdálenost kódu C je > 2. □

Věta o kontrolní matici II

Věta: Každá binární matice s nenulovými a navzájem různými sloupci je kontrolní matice binárního lineárního kódu, který opravuje jednoduché chyby.

Důkaz: Nechť H je matice, která má nenulové a navzájem různé sloupce. Z předchozího slajdu je zřejmé, že žádné slovo b^i a $b^{i,j}$ není kódovým slovem, tedy minimální váha a tím i minimální vzdálenost kódu je > 2. $\square$

Příklad

$$H = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

. . . je kontrolní matice binárního lineárního kódu. Přepíšeme do rovnic:

$$\begin{aligned} (0x_1 + 0x_2 &= 0) \\ x_2 &= 0 \\ x_1 + x_2 &= 0 \end{aligned}$$

Systém má jediné, nulové řešení. Systém tak generuje kód $C = \{00\}$.

Přidáme sloupec. . .

$$H = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

Systém má opět jediné, nulové řešení, takže $C = \{000\}$.

Přidáme ještě jeden sloupec...

$$H = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

- Kódová slova mají délku 4. Jak vypadá C?
- Jak by vypadalo C, kdybychom prohodili poslední dva sloupce?
- Přidáním dalších sloupců zvyšujeme počet informačních symbolů.
- Kolik sloupců může matice nejvýše mít?
- Takový kód opravuje právě jednu chybu. Proč ne dvě?

Kód opravuje jednu chybu

Aby opravoval dvě chyby, musel by mít kód C minimální váhu > 4. Tedy matice musí mít > 4 sloupce. Ovšem pokud má 3 řádky, pak jistě existují tři nebo čtyři sloupce, jejichž součet je nulový sloupec.

Vezmeme libovolné čtyři sloupce i, j, k, l. Protože matice má 3 řádky, hodnost je maximálně 3, tak určitě alespoň jeden ze sloupců je lineárně závislý. Nechť i-tý sloupec je lineárně závislý, tj. $H_i = a_j \cdot H_j + a_k \cdot H_k + a_l \cdot H_l$. Dva nebo tři koeficienty musí být rovné 1 – řekněme, že buď a_j, a_k, nebo a_j, a_k, a_l. Pak součet sloupců i, j, k nebo sloupců i, j, k, l je roven nulovému sloupci.

Což znamená, že pro vektor u s 1 na pozicích i, j, k nebo i, j, k, l platí $Hu^T = \mathbf{0}^T$, tedy u je kódové slovo. Ale váha u je tři, nebo čtyři, stejně tak minimální váha (a tím pádem i vzdálenost) kódu.

Hammingovy kódy

- Perfektní kódy pro opravu jednoduchých chyb.
- Kódová slova mají délku $n = 2^m - 1$, m symbolů je kontrolních. Minimální vzdálenost je 3.
- Nazývají se $(2^m - 1, 2^m - m - 1)$ kódy, např. $(7, 4)$.
- Sloupce kontrolní matice jsou všechny nenulové vektory seřazené lexikograficky (= podle hodnot, pokud bychom je převedli na číslo v desítkové soustavě). Pro $m = 2$:

$$H = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

Definice Hammingova kódu

Definice: Hammingův kód je binární lineární kód, který má, pro nějaké m, $m \times 2^m - 1$ kontrolní matici, jejíž sloupce obsahují všechny nenulové binární vektory.

Příklad kontrolní matice (jedné z mnoha) pro $m = 3$. Dostáváme matici 3×7:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Zakódování

Předchozí matici můžeme přepsat na tento systém rovnic:

$$\begin{aligned} x_4 + x_5 + x_6 + x_7 &= 0 \\ x_2 + x_3 + x_6 + x_7 &= 0 \\ x_1 + x_3 + x_5 + x_7 &= 0 \end{aligned}$$

A to můžeme přepsat na:

$$\begin{aligned} x_5 &= x_2 + x_3 + x_4 \\ x_6 &= x_1 + x_3 + x_4 \\ x_7 &= x_1 + x_2 + x_4 \end{aligned}$$

Vidíme, že x_1, x_2, x_3, x_4 můžeme brát jako informační symboly a x_5, x_6, x_7 jako kontrolní, které dopočítáme.

Dekódování

- Pokud je u kódové slovo, pak $Hu^T = \mathbf{0}^T$.
- Pokud nastane chyba na pozici i, přijmeme slovo $v = u + e^i$.

$$Hv^T = H(u + e^i)^T = Hu^T + He^{iT} = \mathbf{0}^T + He^{iT} = He^{iT} = H_i$$

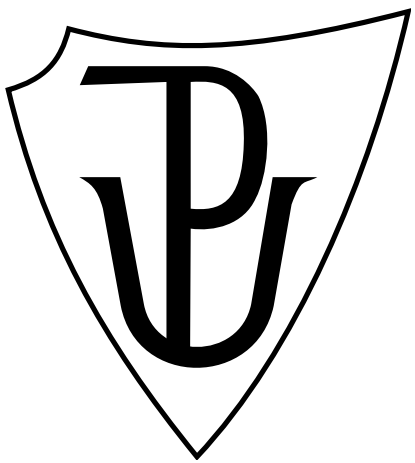
- Součin Hv^T je tak rovný i-tému sloupci matice H.
- V i-tém sloupci je uloženo číslo i v binární podobě.
- $\longrightarrow$ opravíme i-tý znak ve slově v.

Informační poměr

- Informační poměr Hammingova kódu je $R(C) = \frac{2^m - m - 1}{2^m - 1} = 1 - \frac{m}{2^m - 1} \longrightarrow$ malá redundance pro velká m.
- Vždy jsme ale schopni opravit jen jednu chybu.
- Hammingovy kódy jsou perfektní, protože mají nejmenší možnou redundanci na množině kódů opravující jednoduché chyby.
- Každé slovo délky $n = 2^m - 1$ je buď kódové, nebo je od kódového slova vzdáleno o jedna.
- (Během dekódování buď získáme kódové slovo, nebo nám stačí změnit jeden bit.)

Teorie informace a kódování (KMI/TIK)

Lineární kódy



Lukáš Havrlant

Univerzita Palackého

4. prosince 2013

Grupa

- Grupa je algebraická struktura $\mathbf{G} = \langle G, + \rangle$, kde $+$ je binární operace na G splňující:
  - $\forall x, y, z \in G : x + (y + z) = (x + y) + z$ (asociativita)
  - Existuje prvek $0 \in G$ takový, že $\forall x \in G : x + 0 = 0 + x = x$ (neutrální prvek)
  - $\forall x \in G$ existuje $y \in G$ takový, že $x + y = y + x = 0$ (inverzní prvek)
- Pokud navíc $\forall x, y \in G : x + y = y + x$, pak mluvíme o komutativní grupě.
- Příklad: $\mathbb{Z}$ a $\mathbb{R}$ s klasickým sčítáním, $\mathbb{Z}_p = \{0, \ldots, p - 1\}$ s násobením modulo p.

Těleso

- Těleso je algebraická struktura $\mathbf{F} = \langle F, +, \cdot, 0, 1 \rangle$, kde $+$ a $\cdot$ jsou binární operace na F splňující:
  - $\langle F, +, 0 \rangle$ je komutativní grupa,
  - $\langle F \setminus \{0\}, \cdot, 1 \rangle$ je komutativní* grupa,
  - $\forall x, y, z \in G : x \cdot (y + z) = x \cdot y + x \cdot z$ (distributivita).
- Důsledky pro $x, y, z \in F$:
  - $x \cdot 1 = x$, $xy = xz$ implikuje $y = z$.
  - $x + y \in F$ a $xy \in F$. Pokud $x \neq 0, y \neq 0$, pak $xy \neq 0$.
  - Každý prvek x má opačný prvek $-x$ takový, že $x - x = 0$.
  - Každý prvek $x \neq 0$ má inverzní prvek x^{-1} takový, že $xx^{-1} = 1$.
- Příklady: $\mathbb{R}$ s klasickým sčítáním a násobením, $\mathbb{Z}_p$ modulo p , pokud je p prvočíslo.

Vektorový prostor

Definice: Nechť $\mathbf{F} = \langle F, +, \cdot, 0, 1 \rangle$ je těleso. Vektorový prostor na $\mathbf{F}$ je struktura $\mathbf{V}$ obsahující neprázdnou množinu vektorů V, binární operaci $+\colon V \times V \to V$ a binární operaci $\cdot\colon F \times V \to V$ takové, že $\forall a, b \in F, \mathbf{u}, \mathbf{v} \in V$:

- $\langle V, + \rangle$ je komutativní grupa,
- $(ab)\mathbf{v} = a(b\mathbf{v})$,
- $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$,
- $1\mathbf{u} = \mathbf{u}$.

Příklad vektorového prostoru

- Nechť F je těleso. Pak množina $F^n = \{\langle a_1, \ldots, a_n \rangle \mid a_i \in F\}$, kde sčítání $+$ vektorů a násobení $\cdot$ vektoru je definováno klasicky, tvoří vektorový prostor.

- $\mathbb{R}, \mathbb{Z}_p, \ldots$

- Pro $p = 2$ dostáváme u $\mathbb{Z}_p$ vektorový prostor, který jsme použili u binárních lineárních kódů.

Vektorový podprostor

- Nechť V je vektorový prostor na tělese F. Pak $\emptyset \neq U \subseteq V$ s operacemi $+$ a $\cdot$ nazveme vektorový podprostor, pokud platí $\forall \mathbf{u}, \mathbf{v} \in U, a \in F$:
  - $\mathbf{u} + \mathbf{v} \in U$,
  - $a\mathbf{u} \in U$.
- Příklad: Pro těleso F, množina $\{\langle 0, a_2, \ldots, a_n \rangle \mid a_i \in F\}$ je vektorový podprostor prostoru F^n.
- $\{\mathbf{0}\}$ je triviální vektorový podprostor.

Lineární kombinace

- Lineární kombinace vektorů $\mathbf{u}_1, \ldots, \mathbf{u}_n \in V$ je vektor $a_1\mathbf{u}_1 + \cdots + a_n\mathbf{u}_n$, kde $a_1, \ldots, a_n \in F$.
- Pokud máme danou množinu vektorů $\mathbf{u}_1, \ldots, \mathbf{u}_n \in V$, pak množina všech lineárních kombinací těchto vektorů tvoří vektorový podprostor prostoru V.
- Tento podprostor značíme obal$(\{\mathbf{u}_1, \ldots, \mathbf{u}_n\})$.
- obal$(\{\mathbf{u}_1, \ldots, \mathbf{u}_n\})$ je nejmenší podprostor obsahující vektory $\mathbf{u}_1, \ldots, \mathbf{u}_n$.

Nezávislosti vektorů, báze, dimenze

- Množina vektorů $U \subseteq V$ se nazývá lineárně nezávislá, pokud žádný vektor $\mathbf{u} \in U$ není lineární kombinací $U \setminus \{\mathbf{u}\}$.
- Báze prostoru V je lineárně nezávislá množina $U \subseteq V$, pro kterou platí obal$(U) = V$.
- Pokud je U báze prostoru V, pak má prostor V dimenzi $|U|$.
- Pokud $U = \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ je báze prostoru V, pak pro každé $\mathbf{u} \in V$ existují unikátní skaláry $a_1, \ldots, a_n \in F$ tak, že $\mathbf{u} = a_1\mathbf{e}_1 + \cdots + a_n\mathbf{e}_n$.
- Pokud $|F| = r$, pak každý k-rozměrný vektorový prostor na F má r^k prvků.

Lineární kódy

Definice: Nechť F je konečné těleso. Pak lineární (n, k) kód je k-dimenzionální vektorový (lineární) podprostor prostoru F^n.

- Lineární (n, k) kód má k informačních symbolů a $n - k$ kontrolních.
- Nechť C je lineární (n, k) kód. Pak má bázi $\mathbf{e}_1, \ldots, \mathbf{e}_k$.
- Každé slovo $\mathbf{v} \in C$ lze vyjádřit jako lineární kombinace vektorů báze: $\mathbf{v} = \sum_{i=1}^{k} u_i \mathbf{e}_i$. Koeficienty $u_i \in F$ tvoří slovo $\mathbf{u}$ délky k.
- A naopak: každé slovo $\mathbf{u}$ délky k definuje kódové slovo $\mathbf{v}$: $\mathbf{v} = \sum_{i=1}^{k} u_i \mathbf{e}_i$

Příklad

Báze kódu sudé parity délky 4:

$$\mathbf{e}_1 = 1100, \quad \mathbf{e}_2 = 1010, \quad \mathbf{e}_3 = 1001$$

Slovo $\mathbf{v} = 0011$ vyjádříme jako

$$\mathbf{v} = 0 \cdot 1100 + 1 \cdot 1010 + 1 \cdot 1001, \qquad (\mathbf{u} = 011)$$

Slovo $\mathbf{v} = 1010$ jako

$$\mathbf{v} = 0 \cdot 1100 + 1 \cdot 1010 + 0 \cdot 1001, \qquad (\mathbf{u} = 010)$$

Generující matice

Definice:

Pokud je C lineární kód s bází $\mathbf{e}_1, \ldots, \mathbf{e}_k$, pak matice G typu $k \times n$ a tvaru

$$G = \begin{pmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_k \end{pmatrix} = \begin{pmatrix} e_{11} & \ldots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{k1} & \ldots & ekn \end{pmatrix}$$

se nazývá generující matice kódu C.

Příklad generující matice: kód sudé parity

Generující matice kódu sudé parity délky 4:

$$G = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{pmatrix}$$

(Jak by vypadala matice pro kód liché parity?) Co musí platit:

- Každý řádek je kódové slovo.
- Řádky jsou nezávislé.
- Kód má délku 4, matice musí mít 4 sloupce.
- Kód má 1 kontrolní a 3 informační symboly, musí mít 3 řádky.
- Každé kódové slovo lze vyjádřit jako kombinace řádků matice.

Elementární úpravy generující matice

- Pokud je G generující matice kódu C, pak matice G' vzniklá z G pomocí elementárních řádkových maticových úprav je opět generující maticí kódu C.

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Další příklad

- Nechť $F = \mathbb{Z}_3$. Matice

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

  je generující matice lineárního $(6, 3)$ kódu.

- Řádky jsou nezávislé, takže tvoří bázi 3-dimenzionálního podprostoru prostoru F^6.

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

- Kód C obsahuje slova $112200, 220011$, atd.

Kódování

- Nechť G je $k \times n$ generující matice obsahující vektory $\mathbf{e}_1, \ldots, \mathbf{e}_k$.
- Každý vektor $\mathbf{u} = u_1 \ldots u_k$ jednoznačně definuje vektor $\mathbf{v}$: $\mathbf{v} = \sum_{i=1}^{k} u_i \mathbf{e}_i$
- Tedy $\mathbf{v} = \mathbf{u}G$
- Slovo $\mathbf{u}$ tak můžeme zakódovat pomocí předpisu:

$$e(\mathbf{u}) = \mathbf{u}G$$

Systematický kód

- Různé generující matice vedou k různým pravidlům kódování.
- Nejběžnější způsob je systematický kód: vybereme $u_1 \dots u_k$ symbolů a doplníme $u_{k+1} \dots u_n$.
- Generující matice má pak tvar $G = (E|B)$, kde E je jednotková $k \times k$ matice a B je $k \times n - k$ matice.

> **Definice:** Lineární kód se nazývá systematický, pokud má generující matici tvaru $G = (E|B)$.

Příklad generující matice systematického kódu

Generující matice kódu sudé parity délky $n = 4$:

$$\begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Generující matice Hammingova $(7, 4)$ kódu:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Příklad kódování

Zakódujeme slovo $\mathbf{u} = 100$ pomocí kódu sudé parity:

$$\begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$$

Výsledné kódové slovo je $\mathbf{v} = 1001$.

Případně můžeme slovo $\mathbf{v}$ spočítat pomocí koeficientů a vektorů báze:

$$\mathbf{v} = 1 \cdot (1001) + 0 \cdot (0101) + 0 \cdot (0011) = (1001)$$

Ekvivalentní kód

Definice: Kódy C_1 a C_2 se nazývají ekvivalentní, pokud existuje permutace π množiny $\{1, \ldots, n\}$ taková, že pro všechna $u_1, \ldots, u_n \in F$:

$$u_1, \ldots, u_n \in C_1 \text{ právě když } u_{\pi(1)}, \ldots u_{\pi(n)} \in C_2$$

- Hammingův $(7, 4)$ kód je systematický, kód sudé parity délky 4 je systematický. Lineární $(6, 3)$ kód z předchozích slajdů není systematický, ale je ekvivalentní systematickému kódu.

Věta: Každý lineární kód je ekvivalentní systematickému kódu.

Kontrolní matice

Definice: Matice H, která má n sloupců a obsahuje prvky z tělesa F se nazývá kontrolní matice lineárního kódu C, jestliže pro všechna slova $\mathbf{u} \in F^n$ platí

$$\mathbf{u} \in C \text{ právě tehdy když } H\mathbf{u}^T = \mathbf{0}^T.$$

Vztah kontrolní a generující matice

Věta: Pro systematický kód C s generující maticí $G = (E_1|B)$ platí, že matice $H = (-B^T|E_2)$ je kontrolní matice kódu C, kde E_1 je jednotková matice typu $k \times k$ a E_2 je jednotková matice typu $n-k \times n-k$.

Důsledek: Každý lineární (n, k) kód má $n-k \times n$ kontrolní matici o hodnosti $n-k$.

Příklady kontrolní matice kódu sudé parity

Generující matice paritního kódu $n = 4, k = 3$:

$$G = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = (E_1|B)$$

Takže kontrolní matice $H = (-B^T|E_2)$ by vypadala: $-B^T = (111)$, E_2 by byla typu 1×1, tedy (1):

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}$$

Příklad kontrolní matice Hammingova kódu

Pro Hammingův $(7, 4)$ kód máme generující matici:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} = (E_1|B)$$

Kontrolní matice:

$$H = (-B^T|E_2) = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Důkaz

Důkaz: Máme ověřit, že podprostor C generovaný generující maticí G je shodný s podprostorem C' všech řešení rovnice $H\mathbf{v}^T = \mathbf{0}$. Přitom platí:

$$HG^T = \left(-B^T | E_2\right) \cdot \begin{pmatrix} E_1 \\ B^T \end{pmatrix} = -B^T E_1 + E_2 B^T = -B^T + B^T = \mathbf{0}$$

Tedy pokud $G = (E_1|B) = \begin{pmatrix} \mathbf{e}_1 \\ \dots \\ \mathbf{e}_k \end{pmatrix}$, pak $H\mathbf{e}_i^T = \mathbf{0}$ pro všechna i. Z toho vyplývá, že všechna $\mathbf{e}_i$ jsou řešením rovnice $H\mathbf{v}^T = \mathbf{0}$ a že $\{\mathbf{e}_1, \dots, \mathbf{e}_k\} \subseteq C'$. Pokud $\{\mathbf{e}_1, \dots, \mathbf{e}_k\} \subseteq C'$, pak i $\text{obal}(\{\mathbf{e}_1, \dots, \mathbf{e}_k\}) \subseteq C'$, přičemž $\text{obal}(\{\mathbf{e}_1, \dots, \mathbf{e}_k\}) = C$. Prostor C je tak podprostorem prostoru C'.

Pokračování důkazu

$$G = (E_1|B), \quad H = (-B^T|E_2)$$

Nyní můžeme dokázat, že platí i $C \supseteq C'$, ale stačí nám dokázat, že prostory mají stejnou dimenzi, což bude jednodušší.

Matice G a tedy i matice B má k (lineárně nezávislých) řádků. Dimenze prostoru C je tak rovna k.

Jednotková matice E_2 je řádu $n-k$, takže matice H má hodnost $h(H) = n-k$. Odtud plyne, že množina všech řešení rovnice $H\mathbf{v}^T = \mathbf{0}$ tvoří podprostor dimenze $n - h(H) = n - (n-k) = k$.

Dimenze prostorů C a C' jsou stejné, platí $C \subseteq C'$, takže $C = C'$. □

Dekódování: Syndrom

- Opakování: pokud odešleme slovo $\mathbf{u}$ a přijmeme slovo $\mathbf{v}$, pak slovo $\mathbf{e} = \mathbf{v} - \mathbf{u}$ nazýváme chybové slovo.

- Pozor – už neplatí rovnost $\mathbf{v} - \mathbf{u} = \mathbf{v} + \mathbf{u}$.

Definice: Nechť H je $m \times n$ kontrolní matice kódu C. Syndrom slova $\mathbf{v}$ je slovo $\mathbf{s}$ definované jako

$$\mathbf{s}^T = H\mathbf{v}^T.$$

Vztah chybového slova a syndromu

Věta: Pokud vyšleme kódové slovo $\mathbf{u}$, přijmeme nekódové slovo $\mathbf{v}$, tak chybové slovo $\mathbf{e} = \mathbf{v} - \mathbf{u}$ a slovo $\mathbf{v}$ mají stejný syndrom. Tedy platí:

$$H\mathbf{e}^T = H\mathbf{v}^T$$

Důkaz: (využíváme faktu, že $H\mathbf{u}^T = \mathbf{0}^T$)

$$H\mathbf{e}^T = H(\mathbf{v} - \mathbf{u})^T = H\mathbf{v}^T - H\mathbf{u}^T = H\mathbf{v}^T - \mathbf{0}^T = H\mathbf{v}^T$$

Třída slova **e**

- Nechť C je lineární (n, k) kód s kontrolní maticí H, která vznikla z generující matice G.

- Pro všechna slova $\mathbf{e} \in F^n$ definujeme množinu

$$\mathbf{e} + C = \{\mathbf{e} + \mathbf{u} \,|\, \mathbf{u} \in C\},$$

  kterou nazveme třída slova **e**.

- $\mathbf{e} + C$ je množina všech slov, která můžeme přijmout, pokud nastane chyba **e**.

Vlastnosti třídy slov

- Pro libovolná $\mathbf{e}, \mathbf{e}' \in F^n$:
- Pokud $\mathbf{e} \in C$, pak $\mathbf{e} + C = C$.
- Buď $\mathbf{e} + C = \mathbf{e}' + C$, nebo $\mathbf{e} + C \cap \mathbf{e}' + C = \emptyset$.
- Počet slov v každé třídě je roven počtu kódových slov, tj. $|\mathbf{e} + C| = |C|$.
- Všechna slova z $\mathbf{e} + C$ mají stejný syndrom.

Příklad

Binární $(4, 3)$ kód sudé parity délky 3:

$$C = \{000, 101, 110, 011\}$$

Zvolme $\mathbf{e} = 110$:

$$\mathbf{e} + C = \{110, 011, 000, 101\}$$

Zvolme $\mathbf{e}' = 010$:

$$\mathbf{e}' + C = \{010, 111, 100, 001\}$$

Dekódování

- Reprezentant třídy $\mathbf{e} + C$ je slovo $\mathbf{r}$, které má minimální Hammingovu váhu ze všech slov z $\mathbf{e} + C$.
- Obecný princip dekódování pro jakýkoliv lineární kód:
- Přijmeme slovo $\mathbf{v}$. Spočítáme syndrom: $\mathbf{s}^T = H\mathbf{v}^T$.
- Z třídy slov $\mathbf{s} + C$ nalezneme reprezentanta $\mathbf{r}$. (Proč? Předpokládáme, že nastalo co nejméně chyb. Čím menší váhu slovo $\mathbf{r}$ bude mít, tím méně znaků změníme.)
- Dekódujeme na slovo $\mathbf{u} = \mathbf{v} - \mathbf{r}$.

Příklad

Kontrolní kód sudé parity délky 3 neopravuje žádnou chybu. Kontrolní matice:

$$H = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

Vyšleme slovo $\mathbf{u} = 110$, přijeme slovo $\mathbf{v} = 100$. Spočítáme syndrom:

$$\mathbf{s} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T = 1$$

Syndromu $\mathbf{s} = 1$ odpovídá třída $100 + C = \{010, 111, 100, 001\}$. Vybereme reprezentanta $\longrightarrow$ problém, tři slova mají váhu 1. Pokud vybereme $\mathbf{e} = 010$, opravíme správně:

$$\mathbf{u} = \mathbf{v} - \mathbf{e} = 100 - 010 = 110$$

Pokud vybereme $\mathbf{e} = 001$, opravíme špatně:

$$\mathbf{u} = \mathbf{v} - \mathbf{e} = 100 - 001 = 101$$

Poučení z příkladu

- Pokud má lineární kód opravovat chybu, musí ve třídě $\mathbf{e} + C$ existovat vždy jediný unikátní nejmenší reprezentant třídy.

  Věta: Pokud má lineární kód C minimální vzdálenost $d(C)$, pak kód opravuje $t < d(C)/2$ chyb. Pokud došlo během přenosu k t chybám, kterým odpovídá chybové slovo $\mathbf{e}$, tak třída $\mathbf{e}+C$ obsahuje právě jedno slovo o váze $< d(C)/2$ a to slovo $\mathbf{e}$.

Důkaz

Dokážeme, že pokud lineární kód C opravuje t chyb, které reprezentuje chybové slovo $\mathbf{e}$ o váze t, tak třída $\mathbf{e} + C$ obsahuje právě jedno slovo o váze $< d(C)/2$ a to slovo $\mathbf{e}$.

Předpokládejme, že jsme poslali slovo $\mathbf{u}$, přijali slovo $\mathbf{v}$ a během přenosu nastalo t chyb, $\mathbf{e} = \mathbf{v} - \mathbf{u}$ a $|\mathbf{e}| = t$, kde $|\mathbf{e}|$ je váha slova $\mathbf{e}$.

Sporem předpokládejme, že třída $\mathbf{e} + C$ obsahuje různá slova $\mathbf{v}_1$ a $\mathbf{v}_2$ taková, že $|\mathbf{v}_1| < d(C)/2$ a $|\mathbf{v}_2| < d(C)/2$. Pak ale také $\delta(\mathbf{v}_1, \mathbf{v}_2) < d(C)$. Přitom ale $\mathbf{v}_1 = \mathbf{u}_1 + \mathbf{e}$ a $\mathbf{v}_2 = \mathbf{u}_2 + \mathbf{e}$ pro nějaká $\mathbf{u}_1, \mathbf{u}_2 \in C$. Po dosazení máme $\delta(\mathbf{u}_1 + \mathbf{e}, \mathbf{u}_2 + \mathbf{e}) < d(C)$ a tedy i $\delta(\mathbf{u}_1, \mathbf{u}_2) < d(C)$, což je spor s tím, že $\mathbf{u}_1, \mathbf{u}_2 \in C$. Třída $\mathbf{e} + C$ tak obsahuje jediné slovo o váze $< d(C)/2$. Přitom platí, že $\mathbf{0} \in C$, takže $\mathbf{e} + \mathbf{0} = \mathbf{e} \in \mathbf{e} + C$ a $|\mathbf{e}| = t < d(C)/2$. Tedy $\mathbf{e}$ je jediné slovo v třídě $\mathbf{e} + C$, které má váhu menší než $d(C)/2$. □

Příklad

Hammingův $(7, 4)$ kód má kontrolní matici:

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

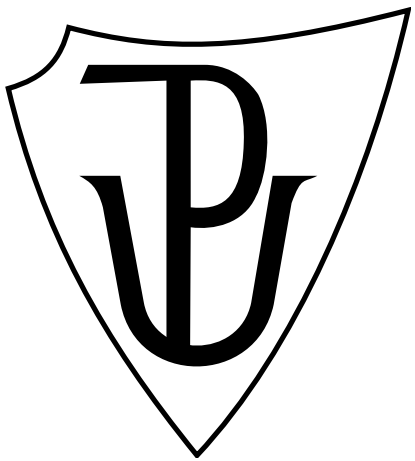
Slovo $\mathbf{e}_0 = 0000000$ má syndrom 000, slovo $\mathbf{e}_1 = 1000000$ má syndrom 001, slovo $\mathbf{e}_2 = 0100000$ má syndrom 010 atd. Dostáváme tak třídy $\mathbf{e}_0 + C, \mathbf{e}_1 + C, \mathbf{e}_2 + C, \ldots$ a reprezentanty $\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2 \ldots$

Pro $\mathbf{v} = 1010111$ máme $H\mathbf{v}^T = (110)^T$. Syndrom 110 má reprezentanta $\mathbf{e}_6 = 0000010$, takže dekódujeme

$$\mathbf{u} = \mathbf{v} - 0000010 = 1010101.$$

Teorie informace a kódování (KMI/TIK)

Reed-Mullerovy kódy



Lukáš Havrlant

Univerzita Palackého

10. ledna 2014

Primární zdroj

Jiří Adámek: Foundations of Coding. Strany 137–160.

Na webu ke stažení, heslo: burzum.

Reed-Mullerovy kódy

- Binární lineární blokový kód, značíme $R(r, m)$.
- Délka kódu je $n = 2^m$
- Počet informačních symbolů: $k = \sum_{i=0}^{r} \binom{n}{i}$
- Minimální vzdálenost: $d = 2^{m-r}$
- Opravuje 2^{m-r-1} chyb
- Pojmenováno po Irving S. Reed a David E. Muller. Muller objevil kód samotný, Reed vymyslel „jednoduchý“ způsob dekódování.
- $R(1, 5)$ je $(32, 6)$ kód, který využil kosmický koráb Mariner 9 při vysílání fotografií z Marsu v roce 1972.

Boolovské funkce

Boolovské funkce

- Boolovská funkce m proměnných je zobrazení $\mathbb{Z}_2^m \longrightarrow \mathbb{Z}_2$.

- Zapisujeme např. pomocí tabulky:

x_1	0	1	0	1
x_2	0	0	1	1
f'	1	0	1	1

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
f''	1	1	0	1	1	1	0	1

- Hodnoty proměnných x_i zapisujeme tak, aby sloupce tvořily čísla $0, 1, \ldots, 2^m - 1$ v binárním zápise.

Reprezentace boolovské funkce

Boolovskou funkci o m proměnných můžeme reprezentovat jako slovo délky 2^m. Např. funkce

x_1	0	1	0	1
x_2	0	0	1	1
f'	1	0	1	1

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
f''	1	1	0	1	1	1	0	1

můžeme reprezentovat jako slova $f' = 1011$ a $f'' = 11011101$, tj. jen jako poslední řádek.

Jakékoliv slovo o délce 2^m reprezentuje nějakou boolovskou funkci.

Definice boolovské funkce

Definice: Slovo $f \in \mathbb{Z}_2^{2^m}$ tvaru $f = f_0 f_1 \dots f_{2^m-1}$, nazveme boolovskou funkcí o m proměnných, definovanou předpisem $f(j_1, j_2, \dots, j_m)$ je rovno f_j, pokud má j binární rozvoj $j_m j_{m-1} \dots j_1$.

Příklad: nechť $f = 11110010$ je boolovská funkce o 3 proměnných. Pak $f(0,0,1)$ vyhodnotíme jako číslo 100_2, což odpovídá číslu 4_{10}, takže $f(0,0,1) = f_4 = 0$.

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
f	1	1	1	1	0	0	1	0

Zajímavé funkce

- Nulová funkce $\mathbf{0} = 0 \ldots 0$
- Funkce $\mathbf{1} = 1 \ldots 1$
- V proměnné x_i se vždy pravidelně střídají skupiny 2^{i-1} nul a 2^{i-1} jedniček.

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
f	1	1	1	1	0	0	1	0

Proměnná jako funkce

- Samotné proměnné x_i jsou také funkcemi. Např. pro $m = 2$ máme $x_1 = 0101$ a $x_2 = 0011$:

x_1	0	1	0	1
x_2	0	0	1	1
f'	0	1	0	1

- Platí, že funkce $f' = x_1$ dvou proměnných je funkce dvou parametrů $f'(x_1, x_2)$, pro kterou platí

$$f'(x_1, x_2) = x_1 = 0101$$

Logické operace

- Použijeme definice operací $+$ a $\cdot$, které jsme použili u binárních lineárních kódů, tj. sčítání a násobení modulo 2.

- Pak pro boolovské funkce $f = \langle f_0, f_1, \ldots, f_{2^m-1} \rangle$ a $g = \langle g_0, g_1, \ldots, g_{2^m-1} \rangle$ a prvek $x \in \mathbb{Z}_2$ definujeme operace:

$$
\begin{aligned}
f \cdot g &= \langle f_0 \cdot g_0, f_1 \cdot g_1, \ldots, f_{2^m-1} \cdot g_{2^m-1} \rangle \\
f + g &= \langle f_0 + g_0, f_1 + g_1, \ldots, f_{2^m-1} + g_{2^m-1} \rangle \\
x \cdot f &= \langle x \cdot f_0, x \cdot f_1, \ldots, x \cdot f_{2^m-1} \rangle \\
x + f &= \langle x + f_0, x + f_1, \ldots, x + f_{2^m-1} \rangle \\
\neg f &= 1 + f \\
f \cdot f &= f
\end{aligned}
$$

Pozorování (Currying)

Pokud máme funkci f o třech proměnných, pak slovo $f_0 f_1 f_2 f_3$ je zároveň funkce dvou proměnných tvaru $f(x_1, x_2, 0)$.

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1
f	0	1	1	1	0	0	1	0

Pokud definujeme $g(x_1, x_2) = f(x_1, x_2, 0)$, pak $g = 0111$, první čtyři symboly funkce f. Podobně pro $f(x_1, x_2, 1)$. Obecně:

$$\begin{aligned} f(x_1, x_2, \ldots, x_{m-1}, 0) &= f_0 f_1 \ldots f_{2^{m-1}-1} \\ f(x_1, x_2, \ldots, x_{m-1}, 1) &= f_{2^{m-1}} f_{2^{m-1}+1} \ldots f_{2^m-1} \end{aligned}$$

Boolovské polynomy

Boolovské polynomy

Boolovské funkce můžeme také reprezentovat jako součty a součiny funkcí x_i a $\mathbf{1}$. Příklad: $f = 0111$.

x_1	0	1	0	1
x_2	0	0	1	1
f	0	1	1	1

Funkci můžeme přepsat takto: $f = x_1 + x_2 + x_1 \cdot x_2$, protože:

$$0101 + 0011 + 0101 \cdot 0011 = 0101 + 0011 + 0001 = 0111$$

Tomuto tvaru říkáme boolovský polynom. Lze každou boolovskou funkci zapsat jako polynom?

Boolovské polynomy 2

- Platí, že $x_i^2 = x_i$, $x_i^1 = x_i$ a $x_i^0 = \mathbf{1}$.

- Boolovský polynom m proměnných je součet funkce $\mathbf{1}$ a členů

$$x_1^{i_1} x_2^{i_2} \dots x_m^{i_m},$$

kde $i_1, \dots, i_m \in \mathbb{Z}_2$. Pro $m = 3$ máme:

$$\begin{aligned} x_1^1 x_2^0 x_3^0 &= x_1 \\ x_1^0 x_2^1 x_3^1 &= x_2 x_3 \\ x_1^0 x_2^0 x_3^0 &= \mathbf{1} \end{aligned}$$

Boolovské polynomy – definice

Definice: Boolovský polynom reprezentující boolovskou funkci f o m proměnných je výraz ve tvaru

$$f = \sum_{i=0}^{2^m-1} q_i \cdot x_1^{i_1} x_2^{i_2} \dots x_m^{i_m},$$

kde $q_i \in \mathbb{Z}_2$ a číslo i má binární rozvoj $i_m i_{m-1} \dots i_2 i_1$.

Pro funkce dvou proměnných tak má polynom tvar

$$q_0\mathbf{1} + q_1x_1 + q_2x_2 + q_3x_1x_2$$

a volbou koeficientů q_i určujeme, jestli v polynomu daný člen „bude“ nebo „nebude“.

Příklad

- Pro $f = x_2 + x_1x_2$ a $m = 2$ máme:

$$q_0 + q_1x_1 + q_2x_2 + q_3x_1x_2,$$

kde $q_0 = q_1 = 0, \quad q_2 = q_3 = 1.$

- Pro $f = 00001111$ a $m = 3$ máme:

$$q_0 + q_1x_1 + q_2x_2 + q_3x_1x_2 + q_4x_3 + q_5x_1x_3 + q_6x_2x_3 + q_7x_1x_2x_3,$$

kde $q_4 = 1$ a zbytek je nula.

Snížení počtu proměnných

Věta: Pro každou boolovskou funkci $m+1$ proměnných $f(x_1, \ldots, x_m, x_{m+1})$ platí

$$f = f(x_1, \ldots, x_m, 0) + [f(x_1, \ldots, x_m, 0) + f(x_1, \ldots, x_m, 1)] \cdot x_{m+1}$$

Důkaz: Dosadíme za x_{m+1} jedna a nula.

Pokud tento postup použijeme rekurzivně dále, můžeme takto získat z boolovské funkce boolovský polynom.

Příklad

Převedeme funkci dvou proměnných $f = 0111$ na polynom.

$$f = 01 + (01 + 11)x_2 = 01 + 10x_2$$

Dále vidíme, že

$$\begin{aligned} 01 &= 0 + (0 + 1)x_1 = x_1 \\ 10 &= 1 + (1 + 0)x_1 = 1 + x_1 \end{aligned}$$

Takže

$$f = 01 + 10x_2 = x_1 + (1 + x_1)x_2 = x_1 + x_2 + x_1x_2.$$

Funkce 0111 odpovídá polynomu $x_1 + x_2 + x_1x_2$.

Důsledek

- Každý boolovský polynom jsme schopni převést na boolovskou funkci. (Zkrátka vše sečteme/vynásobíme.)

- Každou boolovskou funkci jsme schopni převést na boolovský polynom.

Báze lineárního prostoru $\mathbb{Z}_2^{2^m}$

Věta: Mějme vektorový prostor $\mathbb{Z}_2^n$, kde $n = 2^m$. Následující polynomy o jednom sčítanci tvoří bázi B tohoto vektorového prostoru:

$$\mathbf{1},$$
$$x_1, x_2, \ldots, x_m,$$
$$x_i x_j \quad \text{pro} \quad i, j \in \{1, \ldots, m\},$$
$$\vdots,$$
$$x_1 x_2 \ldots x_m$$

Důkaz

Víme, že každá boolovská funkce délky $n = 2^m$ lze vyjádřit jako boolovský polynom o m proměnných. Obalem báze B je zřejmě celý vektorový prostor $\mathbb{Z}_2^n$. Zbývá dokázat, že vektory/polynomy báze jsou nezávislé.

Dokážeme, že počet polynomů v bázi B je shodný s dimenzí prostoru $\mathbb{Z}_2^n$ (dimenze je $n = 2^m$). Máme 1 polynom stupně 0, m polynomů stupně 1, $\binom{m}{2}$ polynomů stupně 2 atd. Celkem máme

$$\binom{m}{0} + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{m-1} + \binom{m}{m} = \sum_{k=0}^{m} \binom{m}{k} = 2^m$$

polynomů. (Rovnost vysvětlena na dalším slajdu.) □

Binomická věta

Proč platí rovnost na předchozím slajdu? Binomická věta říká:

$$(x+y)^m = \sum_{k=0}^{m} \binom{m}{k} x^{m-k} y^k.$$

Pokud dosadíme $x = y = 1$, máme:

$$2^m = \sum_{k=0}^{m} \binom{m}{k} = \binom{m}{0} + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{m-1} + \binom{m}{m}.$$

Stupeň polynomu

Polynom **0** má stupeň -1. Polynom **1** má stupeň 0. Stupeň ostatních boolovských polynomů

$$f = \sum_{i=0}^{2^m-1} q_i \cdot x_1^{i_1} x_2^{i_2} \dots x_m^{i_m}$$

je maximální váha slova $i_1 i_2 \dots i_m$ takového, že $q_i = 1$. Tedy je to největší počet proměnných, které se vyskytují v nějakém sčítanci.

Příklady:

- Polynom $1 + x_1 + x_1 x_2$ má stupeň 2,
- polynom $x_1 x_3 + x_1 x_2 x_3$ má stupeň 3.

Reed-Mullerovy kódy

Reed-Mullerovy kódy

Definice: Reed-Mullerovým kódem stupně r a délky 2^m se nazývá množina $R(r, m)$ všech boolovských polynomů m proměnných stupně nejvýše r.

Kódová slova Reed-Mullerových kódů tak jsou polynomy.

Příklad: Kódy $R(0, m)$ jsou kódy, které obsahují polynomy o stupni maximálně 0, tj. obsahují pouze $\mathbf{0}$ a $\mathbf{1}$. Jedná se tak o opakující kód délky 2^m.

Příklad Reed-Mullerových kódů

18.3. Příklad: Všechny Reedovy-Mullerovy kódy délky 4

0	0 0 0 0	$R(-1, 2)$
1	1 1 1 1	$R(0, 2)$
x_1	0 1 0 1	
x_2	0 0 1 1	
$x_1 + x_2$	0 1 1 0	
$1 + x_1$	1 0 1 0	
$1 + x_2$	1 1 0 0	
$1 + x_1 + x_2$	1 0 0 1	$R(1, 2)$
x_1x_2	0 0 0 1	
$1 + x_1x_2$	1 1 1 0	
$x_1 + x_1x_2$	0 1 0 0	
$x_2 + x_1x_2$	0 0 1 0	
$x_1 + x_2 + x_1x_2$	0 1 1 1	
$1 + x_1 + x_1x_2$	1 0 1 1	
$1 + x_2 + x_1x_2$	1 1 0 1	
$1 + x_1 + x_2 + x_1x_2$	1 0 0 0	$R(2, 2)$

Generující matice

- Reed-Mullerův kód je lineární, protože součet dvou polynomů stupně nejvýše r je také polynom stupně nejvýše r.

- Generující matice tvoří všechny polynomy obsahující jediný výraz ve tvaru součinu o stupni $\leq r$. Matice kódu $R(2,3)$:

$$G = \left[\begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & x_1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & x_2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & x_3 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & x_1x_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & x_1x_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & x_2x_3 \end{array}\right].$$

$R(1,3)$ – Hammingův kód

Podívejme se na generující matici $R(1,3)$:

$$G = \begin{bmatrix} \mathbf{1} \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \left[\begin{array}{c|ccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

Je to generující matice rozšířeného Hammingova $(7,4)$ kódu.

Rozšířený Hammingův kód = ke kódovému slovu Hammingova kódu se přidá jeden symbol tak, aby výsledné kódové slovo mělo sudou paritu.

Odstraněním prvního řádku a sloupce získáme klasický kód.

Kódování

- Slovo, které chceme zakódovat, má délku k, kde

$$k = \sum_{i=0}^{r} \binom{m}{i}$$

- Pokud chceme zakódovat slovo u, pak symboly u_i nám říkají, které součiny z generující matice budou ve výsledném polynomu.

- Dále postupujeme standardně, kódování $e : \mathbb{Z}_2^k \to R(r, m)$ bude probíhat takto:

$$e(u) = u \cdot G$$

Příklad kódování

Použijeme generující matici:

$$G = \left[\begin{array}{cccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & x_1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & x_2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & x_3 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & x_1x_2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & x_1x_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & x_2x_3 \end{array}\right] .$$

Zakódujeme slovo 0010110:

$$0010110 \cdot G = x_2 + x_1x_2 + x_1x_3 = 00100111$$

Vidíme, že kódování odpovídá kódu sudé parity.

Vlastnosti

- $R(m, m) = \mathbb{Z}_2^{2^m}$
- $R(m-1, m)$ je $(2^m, 2^m - 1)$ kód sudé parity (ale není systematický, viz příklad).
- $R(0, m)$ je $(2^m, 1)$ opakovací kód.
- Minimální vzdálenost $R(r, m)$ kódu je 2^{m-r}.

Geometrická interpretace

Analytická geometrie: opakování

Máme Euklidovský prostor $\mathbb{R}^3$, který tvoří vektorový prostor, a body $x_1x_2x_3 \in \mathbb{R}^3$. V $\mathbb{R}^3$ máme přímky popsané parametricky

$$\mathbf{a} + t\mathbf{b},$$

kde $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$ a $t \in \mathbb{R}, \mathbf{b} \neq \mathbf{0}$. Plochu popíšeme jako

$$\mathbf{a} + t\mathbf{b} + s\mathbf{c},$$

kde $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$, $\mathbf{b}, \mathbf{c}$ jsou lineárně nezávislé a $t, s \in \mathbb{R}$.

Binární Euklidovský třídimenzionální prostor

Body tohoto prostoru bude množina $\mathbb{Z}_2^3$. Tzn., že prostor obsahuje **konečně** mnoho bodů! Konkrétně osm. Můžeme je všechny zobrazit do tabulky.

Bod
$\mathbf{p}_0 = 000$
$\mathbf{p}_1 = 001$
$\mathbf{p}_2 = 010$
$\mathbf{p}_3 = 011$
$\vdots$
$\mathbf{p}_7 = 111$

Charakteristická funkce bodu

Každému bodu $\mathbf{p}_i \in \mathbb{Z}_2^3$ přiřadíme charakteristickou funkci $f = f_7 f_6 \ldots f_1 f_0$. Platí, že

$$f(\mathbf{p}_i) = f_7 f_6 \ldots f_1 f_0, \quad \text{kde } f_j = \begin{cases} 1 & \text{pokud } j = i \\ 0 & \text{jinak} \end{cases}$$

Tabulka:

Bod	Charakteristická funkce f
$\mathbf{p}_0 = 000$	00000001
$\mathbf{p}_1 = 001$	00000010
$\mathbf{p}_2 = 010$	00000100
$\mathbf{p}_3 = 011$	00001000
$\vdots$	$\vdots$
$\mathbf{p}_7 = 111$	10000000

Charakteristická funkce množiny

Pokud $P \subseteq \mathbb{Z}_2^3$, pak můžeme definovat charakteristickou funkci f množiny P takto:

$$f(P) = \sum_{\mathbf{p} \in P} \mathbf{p}$$

Jinými slovy, $f(P) = f_7 f_6 \ldots f_1 f_0$, kde $f_i = 1$ právě tehdy, když $\mathbf{p}_i \in P$, jinak $f_i = 0$. Příklad:

$$\begin{aligned} f(\{\mathbf{p}_1, \mathbf{p}_7\}) &= 00000010 + 10000000 = 10000010 \\ f(\emptyset) &= 00000000 \\ f(\mathbb{Z}_2^3) &= 11111111 \end{aligned}$$

Přímka v $\mathbb{Z}_2^3$

Přímka je opět popsána parametricky jako

$$\mathbf{a} + t\mathbf{b},$$

ale $\mathbf{a}, \mathbf{b} \in \mathbb{Z}_2^3$ a $t \in \mathbb{Z}_2, \mathbf{b} \neq \mathbf{0}$. Přímka tak má právě dva body $\mathbf{a}, \mathbf{a} + \mathbf{b}$.

Naopak každá dvojice různých bodů $\mathbf{a}, \mathbf{b}$ tvoří přímku danou předpisem

$$\mathbf{a} + t(\mathbf{b} - \mathbf{a}).$$

Množina všech přímek pak vypadá takto:
$\{\{\mathbf{p}_0, \mathbf{p}_1\}, \{\mathbf{p}_0, \mathbf{p}_2\}, \ldots, \{\mathbf{p}_6, \mathbf{p}_7\}\}$. Celkem jich je

$$\binom{8}{2} = 27.$$

Plocha v $\mathbb{Z}_2^3$

Plocha je také popsána parametricky

$$\mathbf{a} + t_1\mathbf{b}_1 + t_2\mathbf{b}_2,$$

kde $\mathbf{a}, \mathbf{b}_1, \mathbf{b}_2 \in \mathbb{Z}_2^3$, $\mathbf{b}_1, \mathbf{b}_2$ jsou nezávislé a $t_1, t_2 \in \mathbb{Z}_2$. Plocha se tak skládá ze čtyř bodů

$$\mathbf{a}, \quad \mathbf{a} + \mathbf{b}_1, \quad \mathbf{a} + \mathbf{b}_2, \quad \mathbf{a} + \mathbf{b}_1 + \mathbf{b}_2.$$

Všechny čtyři body budou po dvou různé, což vyplývá z nezávislosti $\mathbf{b}_1$ a $\mathbf{b}_2$.

Všechny plochy v $\mathbb{Z}_2^3$

Plane	Characteristic Function	Boolean Polynomial
$\{p_1, p_3, p_5, p_7\}$	10101010	x_0
$\{p_2, p_3, p_6, p_7\}$	11001100	x_1
$\{p_4, p_5, p_6, p_7\}$	11110000	x_2
$\{p_0, p_2, p_4, p_6\}$	01010101	$1 + x_0$
$\{p_0, p_1, p_4, p_5\}$	00110011	$1 + x_1$
$\{p_0, p_1, p_2, p_3\}$	00001111	$1 + x_2$
$\{p_1, p_2, p_5, p_6\}$	01100110	$x_0 + x_1$
$\{p_1, p_3, p_4, p_6\}$	01011010	$x_0 + x_2$
$\{p_2, p_3, p_4, p_5\}$	00111100	$x_1 + x_2$
$\{p_1, p_2, p_4, p_7\}$	10010110	$x_0 + x_1 + x_2$
$\{p_0, p_3, p_4, p_7\}$	10011001	$1 + x_0 + x_1$
$\{p_0, p_2, p_5, p_7\}$	10100101	$1 + x_0 + x_2$
$\{p_0, p_1, p_6, p_7\}$	11000011	$1 + x_1 + x_2$
$\{p_0, p_3, p_5, p_6\}$	01101001	$1 + x_0 + x_1 + x_2$

Plocha jako boolovský polynom

Všimněme si – každá plocha P má svou charakteristickou funkci $f(P)$, což je boolovská funkce. Každá boolovská funkce lze převést na boolovský polynom.

Množina všech ploch tvoří kód $R(1,3)$, protože obsahuje všechny polynomy o 3 proměnných a stupni maximálně 1.

Plocha jako množina řešení rovnice

Všechny plochy procházející počátkem ($\mathbf{a} = \mathbf{0}$) jsou popsány rovnicí

$$h_0 x_0 + h_1 x_1 + h_2 x_2 = 0.$$

Zvolíme koeficienty h_i a dopočítáme souřadnice x_i. Např. pro $h_0 = 0, h_1 = 1$ a $h_2 = 1$ máme rovnici

$$0 \cdot x_0 + x_1 + x_2 = 0.$$

Množina řešení rovnice má tvar

$$\{000, 011, 100, 111\} = \{\mathbf{p}_0, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_7\}.$$

To odpovídá ploše 10011001, neboli $\mathbf{1} + x_0 + x_1$. Dosazením všech kombinací za h_i získáme všechny plochy procházející počátkem.

Plocha jako podprostor

Věta: Plocha P procházející počátkem $\mathbf{0}$ je dvoudimenzionální podprostor prostoru $\mathbb{Z}_2^3$.
Důkaz: Každá plocha P procházející počátkem lze parametricky vyjádřit jako $\mathbf{0} + t_1\mathbf{b}_1 + t_2\mathbf{b}_2$. Obsahuje tak body

$$\mathbf{0}, \quad \mathbf{b}_1, \quad \mathbf{b}_2, \quad \mathbf{b}_1 + \mathbf{b}_2$$

Plocha tak obsahuje nulový vektor. Ověříme uzavřenost na sčítání:

$$\begin{aligned} \mathbf{b}_1 + \mathbf{b}_2 &\in P \\ (\mathbf{b}_1 + \mathbf{b}_2) + \mathbf{b}_1 = \mathbf{b}_2 &\in P \\ (\mathbf{b}_1 + \mathbf{b}_2) + \mathbf{b}_2 = \mathbf{b}_1 &\in P \end{aligned}$$

□

Ostatní plochy

Snadno ověříme, že ostatní plochy lze popsat rovnicí

$$h_0x_0 + h_1x_1 + h_2x_2 = 1$$

Každou plochu tak lze popsat rovnicí

$$h_0x_0 + h_1x_1 + h_2x_2 = c,$$

kde $c \in \mathbb{Z}_2$.

Přímky jako průnik dvou ploch

Věta: Každou přímku $\mathbf{a} + t\mathbf{b}$ lze popsat jako průnik dvou ploch.

Důkaz: Nechť $\mathbf{b}, \mathbf{d}, \mathbf{d}'$ jsou bází prostoru $\mathbb{Z}_2^3$. Dále mějme plochy

$$P_1 : \mathbf{a} + t\mathbf{b} + s\mathbf{d} = \{\mathbf{a}, \mathbf{a}+\mathbf{b}, \mathbf{a}+\mathbf{d}, \mathbf{a}+\mathbf{b}+\mathbf{d}\}$$
$$P_2 : \mathbf{a} + t\mathbf{b} + s\mathbf{d}' = \{\mathbf{a}, \mathbf{a}+\mathbf{b}, \mathbf{a}+\mathbf{d}', \mathbf{a}+\mathbf{b}+\mathbf{d}'\}$$

Protože jsou body $\mathbf{b}, \mathbf{d}$ a $\mathbf{d}'$ nezávislé, platí, že body $\{\mathbf{a}+\mathbf{d}, \mathbf{a}+\mathbf{d}', \mathbf{a}+\mathbf{b}+\mathbf{d}, \mathbf{a}+\mathbf{b}+\mathbf{d}'\}$ jsou po dvou různé. Průnik je tak roven

$$P_1 \cap P_2 = \{\mathbf{a}, \mathbf{a}+\mathbf{b}\} \quad (= \mathbf{a} + t\mathbf{b}).$$

□

Plocha jako coset

Věta: Pro každou plochu $P : \mathbf{a} + t_1\mathbf{b}_1 + t_2\mathbf{b}_2$ existuje dvoudimenzionální vektorový podprostor K prostoru $\mathbb{Z}_2^3$ takový, že množina $\mathbf{a} + K = \{\mathbf{a} + \mathbf{x} \,|\, \mathbf{x} \in K\}$, je rovna ploše P. Množině $\mathbf{a} + K$ říkáme „coset vektorového podprostoru K prostoru $\mathbb{Z}_2^3$“.
Důkaz: Dvoudimenzionální podprostor K je roven nějaké ploše, která prochází počátkem. Plocha K tak bude mít tvar

$$K : \mathbf{0} + t_1\mathbf{b}_1 + t_2\mathbf{b}_2 = \{\mathbf{0}, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_1 + \mathbf{b}_2\}$$

Množina $\mathbf{a} + K$ pak bude vypadat takto:

$$\mathbf{a} + K = \{\mathbf{a}, \mathbf{a} + \mathbf{b}_1, \mathbf{a} + \mathbf{b}_2, \mathbf{a} + \mathbf{b}_1 + \mathbf{b}_2\},$$

což jsou přesně body plochy P. □

Afinní podprostor

O cosetu v předchozí větě řekneme, že se jedná o afinní podprostor dimenze 2. Jiným slovem také *flat*.

Pokud má flat dimenzi s, mluvíme o s-flatu. 3-flat je tak celý prostor $\mathbb{Z}_2^3$, 2-flat jsou plochy, 1-flat jsou přímky a 0-flat jsou body.

Stejně, jako jsme všechny plochy vyjádřili pomocí cosetů podprostoru dimenze 2, můžeme všechny přímky vyjádřit jako cosety podprostoru dimenze 1 atd.

Pokud máme dva flaty L a L', které mají charakteristické funkce f_L a $f_{L'}$ pak jejich průnik $L \cap L'$ je roven součinu $f_L f_{L'}$.

R(1,3) a R(2,3)

Reed-Mullerův kód $R(1,3)$ je roven množině všech charakteristických funkcí všech ploch v prostoru $\mathbb{Z}_2^3$. Viz výčet všech ploch.

Reed-Mullerův kód $R(2,3)$ je roven obalu množině všech ploch a všech přímek. Proč nestačí jen množina ploch a přímek?

Polynom $x_0 + x_1x_2$ je stupně 2 a tak je v $R(2,3)$. Přitom má charakteristickou funkci 01101010. Funkce má čtyři jedničky, což znamená čtyři body. Přímka je přitom tvořena dvěma body. Tj. neexistuje přímka, která by popsala polynom $x_0 + x_1x_2$.

Stačí ale vzít plochu x_0 a přímku danou součinem ploch x_1x_2 jako lineární kombinaci.

Zobecnění prostoru $\mathbb{Z}_2^3$

Nechť $\mathbb{Z}_2^m$ je binární Euklidovský m-dimenzionální prostor, který obsahuje 2^m bodů $p_0, \ldots, p_{2^m-1}$, kde

$$p_0 = 00\ldots00, p_1 = 00\ldots01, \ldots, p_{2^m-1} = 11\ldots11$$

Dále nechť K je vektorový r-dimenzionální podprostor $\mathbb{Z}_2^m$. Každý coset

$$\mathbf{a} + K = \{\mathbf{a} + \mathbf{b} \,|\, \mathbf{b} \in K\}$$

se nazývá r-flat.

Věta: Charakteristická funkce r-flatu je zároveň boolovský polynom stupně $m - r$.

Reed-Mullerovy kódy jako r-flaty

Reed-Mullerův kód $R(r, m)$ můžeme vidět jako obal charakteristických funkcí flatů o dimenzi nejméně $m - r$.

- $R(1, 3)$ jsou s-flaty, kde $s \geq 2$, tedy $s \in \{2, 3\}$, tj. obal ploch a celého prostoru.
- $R(2, 3)$ jsou s-flaty, kde $s \geq 1$, tedy $s \in \{1, 2, 3\}$, tj. obal přímek, ploch a celého prostoru.
- $R(3, 3)$ jsou s-flaty, kde $s \geq 0$, tedy $s \in \{0, 1, 2, 3\}$, tj. obal bodů, přímek, ploch a celého prostoru.

Dekódování

Skalární součin

Skalární součin slov $\mathbf{a} = a_1 \cdots a_n$ a $\mathbf{b} = b_1 \cdots b_n$ je roven

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i \cdot b_i .$$

Algoritmus pro dekódování

První krok: přijeme slovo $\mathbf{w}$, spočítáme paritu všech $(r+1)$-flatů. Řekneme, že flat L má sudou paritu, pokud $\mathbf{w} \cdot L = 0$, jinak má lichou paritu.

Rekurzivní krok: pro všechny $s = r, r-1, \ldots, 0$ spočítáme paritu s-flatu L tak, že řekneme, že L je sudý, pokud je L obsažen ve více sudých $(s+1)$-flatech než v lichých. Jinak je lichý.

Poslední krok: Opravíme i-tý bit slova $\mathbf{w}$ právě tehdy, když 0-flat $\{\mathbf{p}_i\}$ má lichou paritu.