

Předmět KIV/TI - přednáška 7

Bezpečnostní kódy

Ing. Václav Vais, Ph.D.

vais@kiv.zcu.cz

Modelové důsledky šumu

- Záměna vyslaného znaku za jiný znak
 - možné příčiny – vnější elmg. rušení při přenosu dat po metalických spojích, tepelný šum v elektronických součástkách, kosmické záření,
- Porušení synchronizace (tj. ztráta znaku nebo vytvoření falešného znaku)
 - možné příčiny – chyba, resp. nesprávná funkce reálných komponent přenosového kanálu (např. síťového adaptéru PC); tyto stavy jsou obvykle rychle diagnostikovány a opraveny; vnímáme je jako „mimořádnou událost“, řeší se jinými nástroji než kódováním

Redundance v přirozeném jazyce

- Čeština
 - vyslané slovo OPAKOVÁNO; přijato OPAK**K**VÁNO
 - lze opravit jediným způsobem na OPAKOVÁNO
 - (je to jediné „kódové slovo“, které se od přijatého liší v jediném znaku, „nejbližší slovo“)
 - vyslané slovo OPAKOVÁNO; přijato OPAKOVÁN**R**
 - OPAKOVÁNO, OPAKOVÁNÍ, OPAKOVÁNA, OPAKOVÁNI, OPAKOVÁNY, ... ?
 - nelze opravit, nejbližší „kódové slovo“ není jednoznačné
- V případě „umělých“ kódů můžeme vhodnou konstrukcí umožnit odhalení, resp. (někdy i) opravu. Cesta = zvýšení redundance v kódu.

Motivační příklad

Motivační příklad: Binární kódování hexadecimálních čísel.

$Z = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$ abeceda zdroje

$X = \{0,1\}$ abeceda kanálu

Redundance obou kódů:

$$\rho(K_1) = 0$$

$$\rho(K_2) = 1 - \frac{\log_2 16}{\log_2 32} = 1 - \frac{4}{5} = \frac{1}{5}$$

| Zdrojový znak | Kódová značka | |
|---------------|---------------|-------|
| | Kód 1 | Kód 2 |
| 0 | 0000 | 00000 |
| 1 | 0001 | 00011 |
| 2 | 0010 | 00101 |
| 3 | 0011 | 00110 |
| 4 | 0100 | 01001 |
| 5 | 0101 | 01010 |
| 6 | 0110 | 01100 |
| 7 | 0111 | 01111 |
| 8 | 1000 | 10001 |
| 9 | 1001 | 10010 |
| A | 1010 | 10100 |
| B | 1011 | 10111 |
| C | 1100 | 11000 |
| D | 1101 | 11011 |
| E | 1110 | 11101 |
| F | 1111 | 11110 |

Motivační příklad

- Kód 1 – značky délky 4, kódování využívá všech 16 čtveřic, tedy při jakékoli chybě při přenosu kódové značky je přijata (jiná) kódová značka, příjemce chybu není schopen detekovat
- Kód 2 – značky délky 5, kódování nevyužívá všech 32 pětic; „vkládá redundanci“
 - existuje rozklad množiny všech pětic na dvě třídy
 - *množinu kódových značek*
 - *množinu nekódových kombinací*
 - společná vlastnost všech kódových značek – sudý počet znaků 1 ve značce
 - dojde-li při přenosu značky k chybě v jednom prvku, přijme příjemce nekódovou kombinaci s lichým počtem znaků 1, kód detekuje jednoduché chyby

Motivační příklad

| Zdrojový znak | značka |
|---------------|------------------|
| | POSTNET 74910 |
| 0 | 11000 |
| 1 | 00011 |
| 2 | 00101 |
| 3 | 00110 |
| 4 | 01001 |
| 5 | 01010 |
| 6 | 01100 |
| 7 | 10001 |
| 8 | 10010 |
| 9 | 10100 |

- Kód „dva z pěti.“
- Každé kódové slovo obsahuje právě dva znaky 1.
- Rozpozná jednoduché chyby (jedna nebo tři 1).
- Příklady dvojité chyby ?
- odesláno 100**10** , přijato 10001 ($\in K$)
- odesláno 1**00**10 , přijato 11110 ($\notin K$)

Redundance kódu:

$$\rho(K) = 1 - \frac{\log_2 10}{\log_2 32} = 1 - \frac{3,32}{5} = 0,34$$

Detekce chyb

Předpokládejme blokový kód K délky n nad kódovou abecedou T , tedy

$K \subseteq T^n$, kde

$T^n = \{t_1 t_2 \dots t_n \mid t_i \in T \ \forall i = 1, 2, \dots, n\}$ je množina všech slov délky n nad T

Pak existuje dvoublokový rozklad množiny T^n na *kódové značky* K a *nekódové kombinace* $T^n - K$.

Kodér na straně zdroje dat do sdělovacího kanálu vkládá kódové značky z množiny K , dekodér na straně příjemce z kanálu přijímá obecně slova z T^n .

Přijetí nekódové kombinace, tedy slova z $T^n - K$ znamená, že při přenosu došlo k chybě, jinak řečeno - příjemce *detekuje chybu*.

Přijetí kódové kombinace, tedy slova z K znamená přesně to, že **bud'** při přenosu nedošlo k chybě, **nebo** že došlo k takové chybě, kterou dekodér příjemce není schopen detekovat.

Detekce chyb

t-násobnou chybou rozumíme (libovolnou) chybu, kde je počet chybně přenesených prvků **menší nebo roven t** .

Kód K detekuje t -násobnou chybu právě tehdy, jestliže je při vyslání **libovolné** kódové značky a **libovolné** t -násobné chybě přijata **vždy** nekódová kombinace.

Důsledkem výše uvedené interpretace pojmu t - násobná chyba je fakt, že např. z tvrzení „kód K detekuje čtyřnásobné chyby“ plyne, že detekuje i všechny chyby nižší násobnosti, tedy trojitě, dvojitě a jednoduché. Jinak řečeno – pokud kód nedetekuje např. dvojnásobné chyby, nemůže detekovat ani žádné chyby vyšší násobnosti.

Hammingova vzdálenost

Hammingovou vzdáleností slov u a v (stejné délky n) se rozumí počet pozic, v nichž se slova u a v liší. Formálně:

$$u = u_1 u_2 \dots u_n$$

$$v = v_1 v_2 \dots v_n$$

$$d(u, v) = \text{card} \{i \mid u_i \neq v_i, i = 1, 2, \dots, n\}$$

Poznámka: Symbol card představuje mohutnost množiny, v našem případě mohutnost konečné množiny, tedy počet jejích prvků.

Minimální Hammingovou vzdáleností kódu K se rozumí nejmenší Hammingova vzdálenost mezi dvěma různými kódovými značkami kódu K . Formálně:

$$d_0(K) = \min_{u, v \in K, u \neq v} d(u, v)$$

Je zřejmé, že o tom, jaké chyby je schopen kód detekovat, rozhoduje minimální Hammingova vzdálenost kódu.

Hammingova vzdálenost – ilustrační příklad

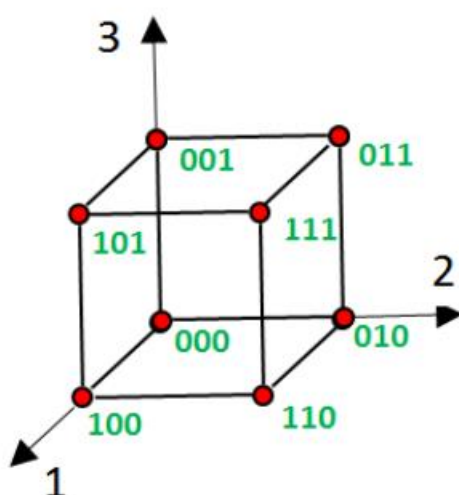
Ilustrační příklad: Binární kódy vybrané z $\{0,1\}^3$ („navlékání kódu na krychli“).

Budeme se zabývat třemi binárními kódy zadanými množinami kódových značek:

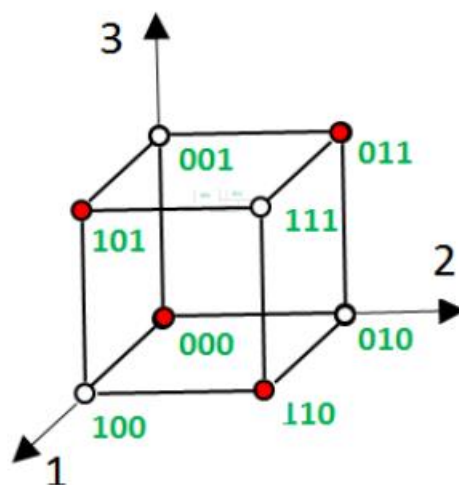
$$K_1 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

$$K_2 = \{000, 011, 101, 110\}$$

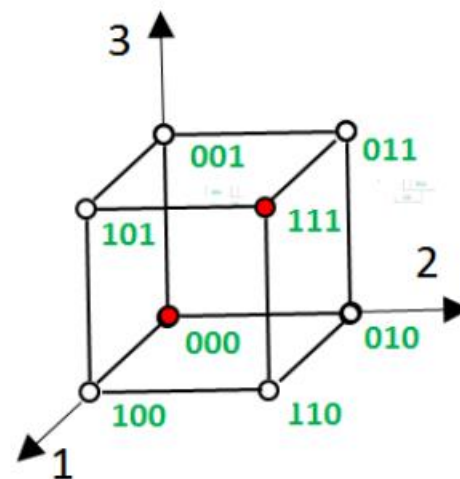
$$K_3 = \{000, 111\}$$



K_1

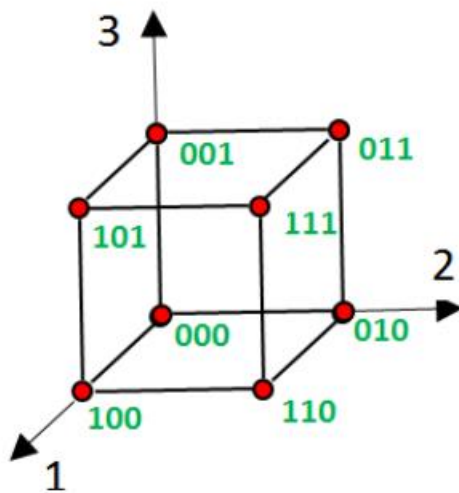


K_2

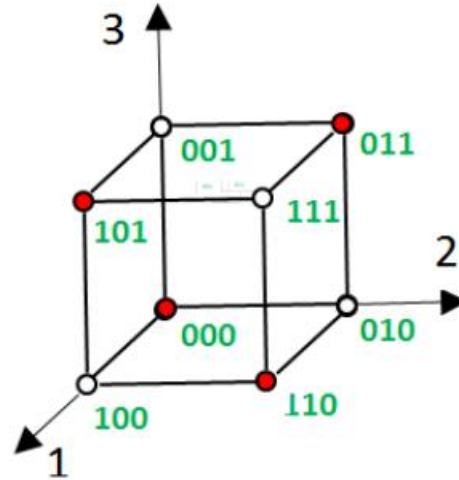


K_3

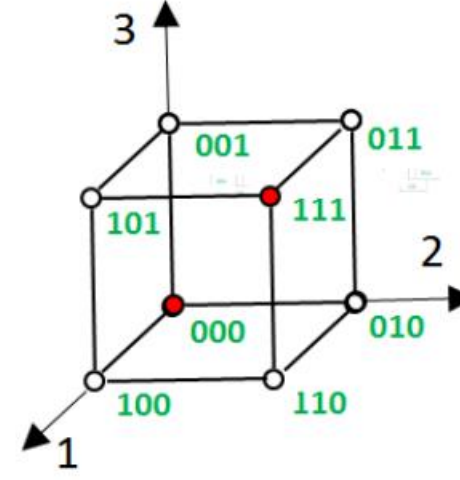
Hammingova vzdálenost – ilustrační příklad



K_1



K_2



K_3

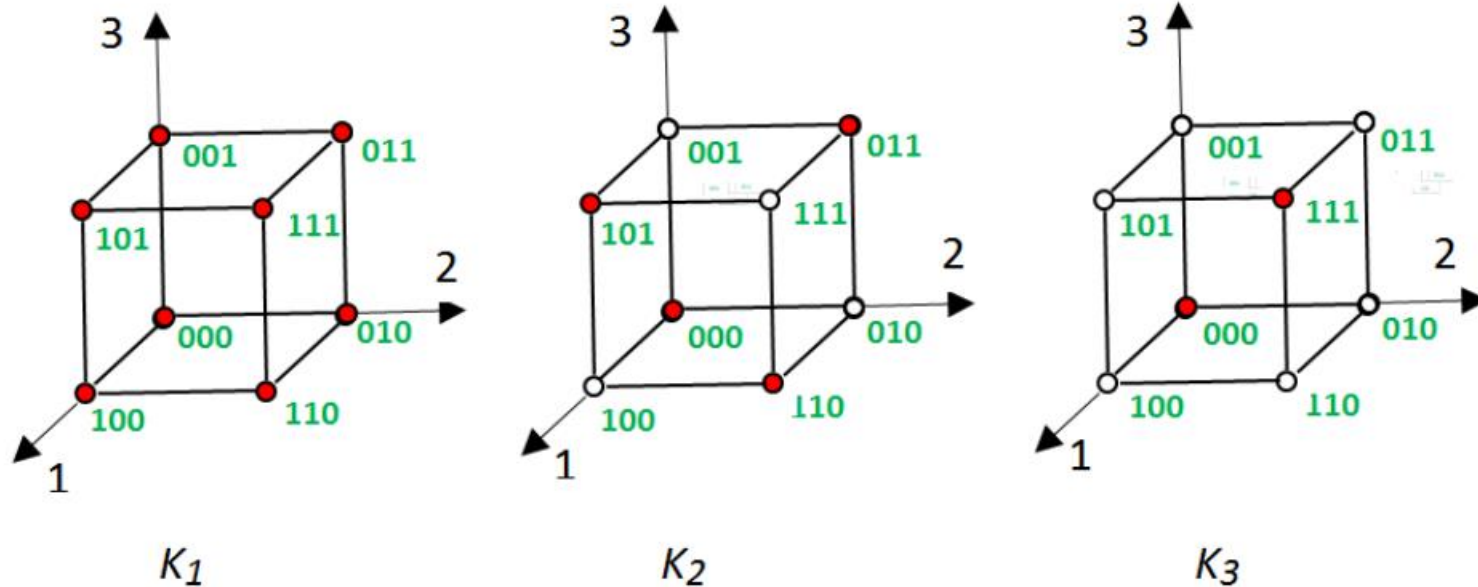
Z obrázků jsou zřejmé minimální Hammingovy vzdálenosti všech tří kódů:

$$d_0(K_1) = 1$$

$$d_0(K_2) = 2$$

$$d_0(K_3) = 3$$

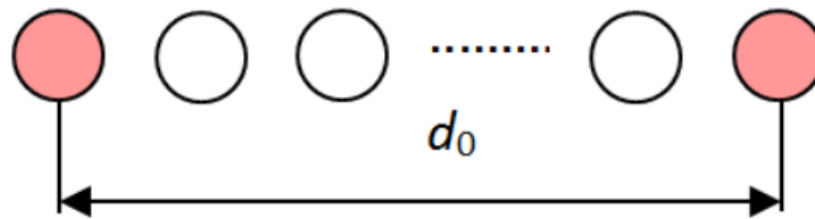
Hammingova vzdálenost – ilustrační příklad



- Kód K_1 nedetekuje žádné chyby (všech osm kombinací je využito jako kódové značky)
- Kód K_2 detekuje jednoduché chyby (neexistují dvě kódové značky, které by se lišily jen v jednom prvku, tj. ležely na jedné hraně krychle; při vyslání libovolné kódové značky a jednoduché chybě v libovolné pozici vždy vznikne nekódová kombinace)
- Kód K_3 detekuje dvojité chyby (Hammingova vzdálenost mezi jedinými dvěma značkami je 3; trojitá chyba už by z kódové značky vytvořila jinou kódovou značku)

Hammingova vzdálenost

- Standardní znázorňování - „nejužší místo“ bezpečnostního kódu, tj. značky, které určují minimální vzdálenost v kódu a všechny nekódové kombinace mezi nimi:



- Zřejmé: Blokový kód s minimální Hammingovou vzdáleností d_0 detekuje všechny chyby s násobností $t < d_0$.

Hammingova vzdálenost - příklad

- Kód celkové kontroly parity (k binárnímu slovu přidáme jeden kontrolní znak tak, aby celkový počet jedniček byl sudý)
- Pro délku $n = 4$ (délka se rozumí včetně paritního prvku):
$$K = \{0000, 1100, 1010, 1001, 0110, 0101, 0011, 1111\}$$
- Minimální Hammingova vzdálenost kódu: $d_0 = 2$
- Kód celkové kontroly parity detekuje jednoduché chyby.

Hammingova vzdálenost - příklad

- Opakovací kód délky n (přenášený znak kódové abecedy n - krát zopakujeme).
- Pro délku $n = 6$ (délkou se rozumí celkový počet znaků ve značce, tj. „informačních“ i „zabezpečovacích“):

$$K = \{000000, 111111\}$$

- Minimální Hammingova vzdálenost kódu: $d_0 = 6$
- Kód celkové kontroly parity s $n = 6$ detekuje pětinasobné chyby.
- Pro obecnou délku n :
 - $d_0 = n$; detekuje $n - 1$ – násobné chyby.

Opravování chyb

- Za určitých předpokladů lze chyby nejen detekovat, ale i opravovat.
- Předpoklady:
 - symetrický binární sdělovací kanál s bitovou chybovostí (Bit Error Rate) p (p je p-st, že při přenosu jednoho konkrétního prvku došlo k chybě; na reálných přenosových kanálech závisí hlavně na *odstupu signálu od šumu*; metalické spoje – $p \approx 10^{-8}$, optické spoje - $p \approx 10^{-10} - 10^{-12}$)
 - statisticky nezávislé chyby (to, zda se v i -tém přenášeném znaku objeví chyby, nezávisí na tom, zda se objeví v jiných přenášených prvcích)

Ilustrace korektnosti opravování chyb

$p_0, p_1, p_2, \dots, p_n$ pravděpodobnosti toho, že při přenosu značky délky n dojde k bezchybnému přenosu, k jednoduché chybě, dvojité chybě atd., až k chybě ve všech n prvcích.

pravděpodobnost bezchybného přenosu p_0 : $p_0 = (1 - p)^n$

pravděpodobnost jednoduché chyby **v jednom konkrétním (i -tém) bitu**: $p \cdot (1 - p)^{n-1}$

pravděpodobnost chybného přenosu jednoho bitu ve značce p_1 : $p_1 = n \cdot p \cdot (1 - p)^{n-1}$

pravděpodobnost dvojité chyby **v konkrétní dvojici bitů (i, j)**: $p^2 \cdot (1 - p)^{n-2}$

pravděpodobnost chybného přenosu dvou bitů ve značce p_2 : $p_2 = \binom{n}{2} \cdot p^2 \cdot (1 - p)^{n-2}$

pravděpodobnost chyby s násobností t : $p_t = \binom{n}{t} \cdot p^t \cdot (1 - p)^{n-t}$

Ilustrace korektnosti opravování chyb

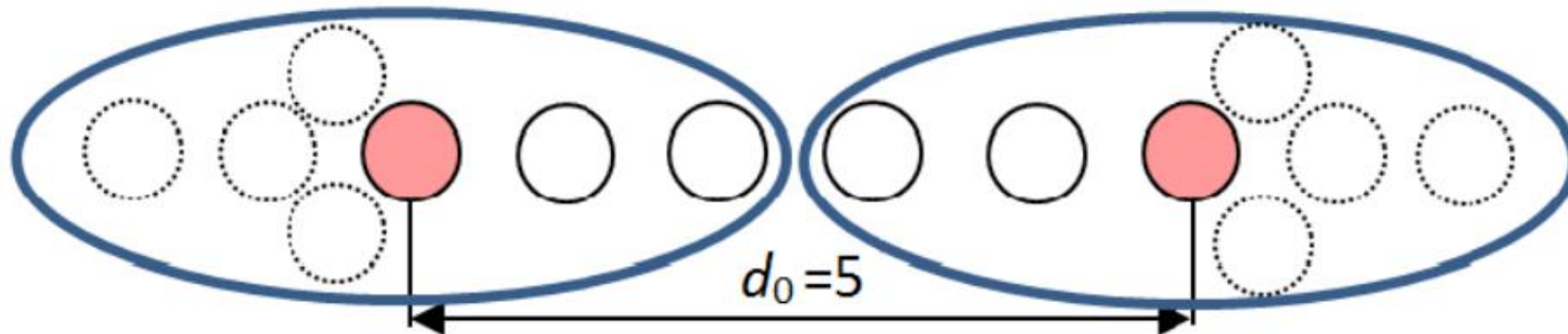
| Násobnost chyby t | bitová chybovost p | | | |
|------------------------|----------------------|------------|------------|------------|
| | 1,0E-06 | 1,0E-08 | 1,0E-10 | 1,0E-12 |
| 0 | 0,99999200 | 0,99999992 | 1,00000000 | 1,00000000 |
| 1 | 0,00000800 | 0,00000001 | 1,0000E-10 | 1,0000E-12 |
| 2 | 2,8000E-11 | 1,0000E-16 | 1,0000E-20 | 1,0000E-24 |
| 3 | 5,6000E-17 | 1,0000E-24 | 1,0000E-30 | 1,0000E-36 |
| 4 | 7,0000E-23 | 1,0000E-32 | 1,0000E-40 | 1,0000E-48 |
| 5 | 5,6000E-29 | 1,0000E-40 | 1,0000E-50 | 1,0000E-60 |
| 6 | 2,8000E-35 | 1,0000E-48 | 1,0000E-60 | 1,0000E-72 |
| 7 | 8,0000E-42 | 1,0000E-56 | 1,0000E-70 | 1,0000E-84 |
| 8 | 1,0000E-48 | 1,0000E-64 | 1,0000E-80 | 1,0000E-96 |

- Tabulka ilustruje oprávněnost opravování chyb na principu hledání „nejbližší kódové značky“ ve smyslu Hammingovy vzdálenosti.

Opravování chyb

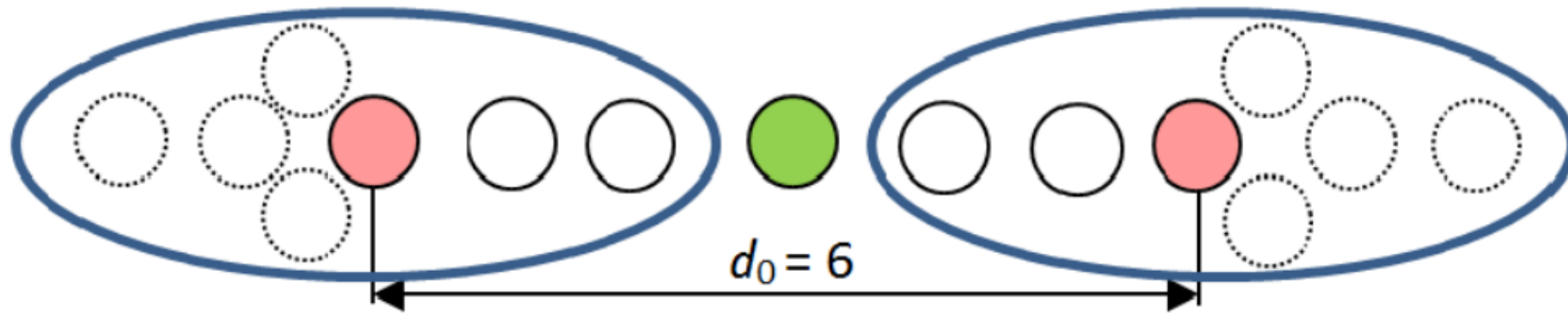
Kód K opravuje t -násobnou chybu právě tehdy, jestliže při vyslání **libovolné** kódové značky $v \in K$ a při **libovolné** t -násobné chybě má přijaté slovo $w \in T^n$ Hammingovu vzdálenost od vyslané kódové značky v menší, než od libovolné jiné kódové značky, platí tedy $\forall x \in K: x \neq v \Rightarrow d(v, w) < d(x, w)$.

Graficky lze opravování na principu nalezení „nejbližší“ kódové značky znázornit tako:



Opravování chyb

Ne vždy lze všem nekódovým kombinacím přiřadit jednoznačně kódové značky :



Zobecnění:

Blokový kód K s minimální Hammingovou vzdáleností d_0 opravuje všechny chyby s násobností $t < d_0 / 2$. Jinak řečeno – má-li kód opravovat t – násobné chyby, musí mít minimální Hammingovskou vzdálenost $d_0 \geq 2 \cdot t + 1$.

Opravování chyb

Zobecnění:

Blokový kód K s minimální Hammingovou vzdáleností d_0 opravuje všechny chyby s násobností $t < d_0 / 2$. Jinak řečeno – má-li kód opravovat t – násobné chyby, musí mít minimální Hammingovskou vzdálenost $d_0 \geq 2 \cdot t + 1$.

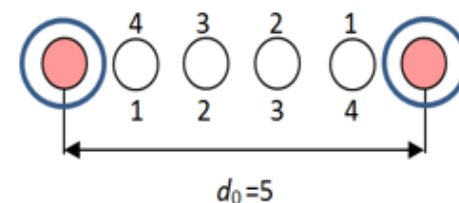
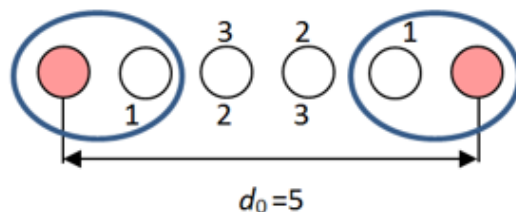
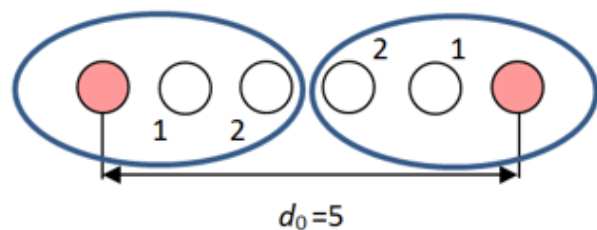
Důsledek:

Opakovací kód **liché** délky n opravuje všechny chyby s násobností $t \leq (n - 1)/2$, opakovací kód **sudé** délky n opravuje všechny chyby s násobností $t \leq (n - 2)/2$.

Opravování chyb

Poznámka: Výše popsáný způsob opravování chyb je založen na tom, že nepředpokládáme výskyt chyb s vyšší násobností, než opravujeme. Pokud by při přenosu došlo k porušení většího počtu bitů než opravujeme, mohlo by dojít k „chybné opravě“ na jinou kódovou značku, než byla odeslána.

Smíšené strategie: Příjemce může zvolit strategii, kdy neopravuje všechny chyby, které mu minimální Hammingova vzdálenost umožňuje. Tím zvýší počet chyb, které je schopen detekovat. Příklad různých strategií pro kód s minimální vzdáleností $d_0 = 5$:



Dekódování s opravou chyb

Dekódováním s opravou t – násobných chyb budeme rozumět parciální funkci

$\delta : T^n \rightarrow K$ takovou, že $\delta(w) = v \Leftrightarrow d(w, v) \leq t$.

Samozřejmá vlastnost funkce δ : $\delta(v) = v \quad \forall v \in K$ (dekódování nemění kódové značky)

Dekódovací funkce δ jednoznačně definuje rozklad množiny všech n – tic T^n :

$T^n = \{K_1, K_2, \dots, K_{\text{card } K}, K_D\}$, kde

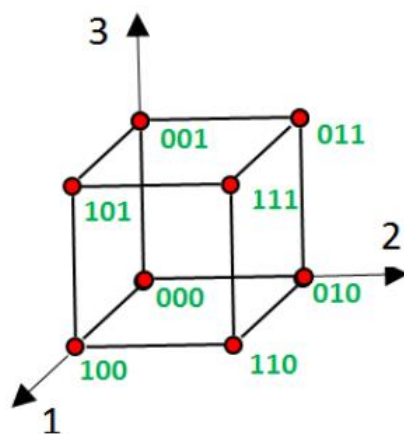
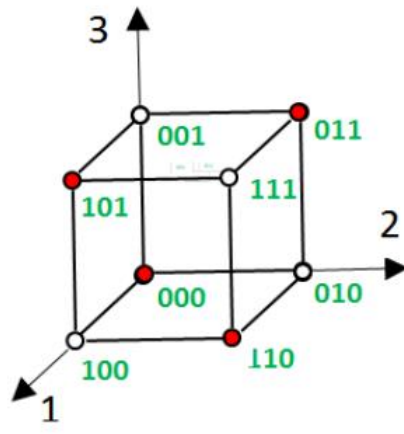
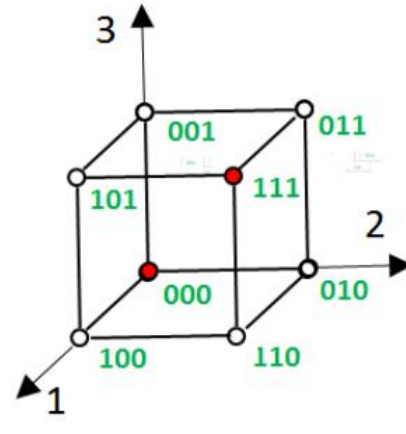
$\text{card } K$ je počet kódových značek

$K = \{v_1, v_2, \dots, v_{\text{card } K}\}$ je množina kódových značek

K_i je množina všech n –tic, které se opravují na kódovou značku v_i

$(\forall i = 1, 2, \dots, \text{card } K)$

K_D je množina všech n – tic, které pouze detekují chybu a neopravují se


 K_1

 K_2

 K_3

| Přijatá trojice w | | Dekódovací funkce δ | | | | | |
|---------------------|-----|----------------------------|-----|-------|-----|-------|-----|
| | | Kód 1 | | Kód 2 | | Kód 3 | |
| w_1 | 000 | v_1 | 000 | v_1 | 000 | v_1 | 000 |
| w_2 | 001 | v_2 | 001 | | X | | 000 |
| w_3 | 010 | v_3 | 010 | | X | | 000 |
| w_4 | 011 | v_4 | 011 | v_2 | 011 | | 111 |
| w_5 | 100 | v_5 | 100 | | X | | 000 |
| w_6 | 101 | v_6 | 101 | v_3 | 101 | | 111 |
| w_7 | 110 | v_7 | 110 | v_4 | 110 | | 111 |
| w_8 | 111 | v_8 | 111 | | X | v_2 | 111 |

| Přijatá trojice w | | Dekódovací funkce δ | | | | | |
|---------------------|-----|----------------------------|-----|-------|-----|-------|-----|
| | | Kód 1 | | Kód 2 | | Kód 3 | |
| w_1 | 000 | v_1 | 000 | v_1 | 000 | v_1 | 000 |
| w_2 | 001 | v_2 | 001 | | X | | 000 |
| w_3 | 010 | v_3 | 010 | | X | | 000 |
| w_4 | 011 | v_4 | 011 | v_2 | 011 | | 111 |
| w_5 | 100 | v_5 | 100 | | X | | 000 |
| w_6 | 101 | v_6 | 101 | v_3 | 101 | | 111 |
| w_7 | 110 | v_7 | 110 | v_4 | 110 | | 111 |
| w_8 | 111 | v_8 | 111 | | X | v_2 | 111 |

Kód 1:

Každá značka je samostatnou třídou rozkladu, $K_i = \{w_i\} \forall i = 1, 2, \dots, \text{card } K$, $K_D = \emptyset$.

Kód 2:

$K_1 = \{000\}$, $K_2 = \{011\}$, $K_3 = \{101\}$, $K_4 = \{110\}$, $K_D = \{001, 010, 100, 111\}$

Kód 3:

$K_1 = \{000, 001, 010, 100\}$, $K_2 = \{011, 101, 110, 111\}$, $K_D = \emptyset$.

Algoritmus dekódování s opravou t -násobných chyb na principu nalezení „nejbližší“ kódové značky

Vstupy algoritmu: přijatá n – tice $\mathbf{w} \in T^n$, množina kódových značek K , násobnost opravených chyb t ($t < d_0(K) / 2$)

Výstup algoritmu: kódová značka $\mathbf{v} \in K$ „nejbližší“ přijaté n – tici \mathbf{w} (pokud existuje), taková, že $d(\mathbf{w}, \mathbf{v}) \leq t$ a je minimální (tj. neexistuje jiná kódová značka $\tilde{\mathbf{v}} \in K$, pro kterou by platilo $d(\mathbf{w}, \tilde{\mathbf{v}}) \leq d(\mathbf{w}, \mathbf{v})$)

Algoritmus:

1. Pro k od 0 do t provedeme kroky 2 a 3
2. Vytvoříme množinu $O_k \subseteq T^n$ jako množinu všech n – tic $\mathbf{u} \in T^n$ takových, že $d(\mathbf{u}, \mathbf{w}) = k$.
3. Pokud je prvkem množiny O_k kódová značka (tj. $O_k \cap K = \{\mathbf{v}\}$), je značka \mathbf{v} výstupem algoritmu, algoritmus končí.
4. Hledaná kódová značka neexistuje, algoritmus končí.

Dekódování s opravou chyb

- Dekódování s opravou chyb není nic jiného, než rozhodnutí, zda $\mathbf{w} \in T^n$ je či není prvkem nějaké množiny (buď K , nebo některé třídy rozkladu K_i)
- Možné přístupy k realizaci dekodování s opravou chyb:
 - algoritmus (viz předchozí slajd)
 - „prokládané“ prohledávání tabulek, v nichž jsou prvky jednotlivých tříd rozkladu seřazeny podle svých pravděpodobností
- Nevýhody obou přístupů:
 - relativní časová náročnost, nestejná počty kroků vyhledávacích operací
- Otázku **Je x prvkem množiny A ?** lze ale (někdy) řešit s konstantní délkou zpracování.

Motivační příklad

Rovina ρ procházející počátkem soustavy souřadnic O je určena normálovým vektorem \vec{n} . Dále je dán bod A . Rozhodněte, zda bod A leží v rovině ρ .

Řešení:

Vytvoříme vektor \vec{a} jako průvodič bodu A , tedy $\vec{a} = A - O$.

Pokud bod A leží v rovině ρ , je jeho průvodič \vec{a} kolmý na \vec{n} skalární součin $\vec{a} \cdot \vec{n}$ tedy bude nulový.

Pokud bod A v rovině ρ neleží, svírá průvodič s \vec{n} jiný úhel než pravý, skalární součin $\vec{a} \cdot \vec{n}$ je proto nenulový.

Platí tedy $A \in \rho \Leftrightarrow a_x \cdot n_x + a_y \cdot n_y + a_z \cdot n_z = 0$

Motivační příklad - závěry

- Zobecnění
 - množina všech tříprvkových vektorů tvoří lineární prostor \mathbf{R}^3 dimenze 3
 - rovina ρ procházející počátkem určuje lineární podprostor $\mathbf{L} \subseteq \mathbf{R}^3$ dimenze 2
 - k podprostoru \mathbf{L} existuje ortogonální doplněk $\bar{\mathbf{L}} \subseteq \mathbf{R}^3$ dimenze 1 (v $\bar{\mathbf{L}}$ leží všechny vektory kolmé na vektory ležící v \mathbf{L} , tj. na průvodiče všech bodů z ρ)
 - vektor z \mathbf{R}^3 leží v \mathbf{L} právě tehdy, je-li kolmý na všechny prvky báze $\bar{\mathbf{L}}$
- Závěr
 - Pokud bychom konstruovali množiny kódových značek jako lineární prostory, mohli bychom otázku **Platí $\mathbf{w} \in \mathbf{K}$?** řešit „výpočetně“ v konstantním čase.

Připomenutí z lineární algebry

T je těleso

$\forall a, b \in T: \exists a + b \in T$ (ke každým dvěma prvkům existuje součet, je určen jednoznačně)

$\forall a, b \in T: \exists a \cdot b \in T$ (ke každým dvěma prvkům existuje součin, je určen jednoznačně)

$\forall a, b, c \in T: (a + b) + c = a + (b + c)$ (asociativnost operace sečítání)

$\forall a, b, c \in T: (a \cdot b) \cdot c = a \cdot (b \cdot c)$ (asociativnost operace násobení)

$\forall a, b \in T: (a + b) = (b + a)$ (komutativnost operace sečítání)

$\forall a, b \in T: (a \cdot b) = (b \cdot a)$ (komutativnost operace násobení)

$\forall a, b, c \in T: a \cdot (b + c) = a \cdot b + a \cdot c$ (distributivnost násobení vzhledem ke sčítání)

$\exists 0 \in T \forall a \in T: a + 0 = a$ (0 = neutrální prvek vzhledem k sečítání; je určen jednoznačně)

$\exists 1 \in T \forall a \in T: a \cdot 1 = a$ (1 = neutrální prvek vzhledem k násobení; určen jednoznačně)

$\forall a \in T \exists -a \in T: a + (-a) = 0$ (opačné prvky; jsou určeny jednoznačně)

$\forall a \in T, a \neq 0 \exists a^{-1} \in T: a \cdot a^{-1} = 1$ (inverzní prvky; jsou určeny jednoznačně)

Připomenutí z lineární algebry

$L \subseteq T^n$ je **lineární prostor**

$\forall \mathbf{a}, \mathbf{b} \in \mathbf{L}: \exists \mathbf{a} \oplus \mathbf{b} \in \mathbf{L}$ (ke každým dvěma prvkům v \mathbf{L} existuje součet; a to jednoznačně)

$\forall \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbf{L}: (\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c} = \mathbf{a} \oplus (\mathbf{b} \oplus \mathbf{c})$ (asociativnost operace sečítání)

$\forall \mathbf{a}, \mathbf{b} \in \mathbf{L}: (\mathbf{a} \oplus \mathbf{b}) = (\mathbf{b} \oplus \mathbf{a})$ (komutativnost operace sečítání)

$\exists \mathbf{0} \in \mathbf{L} \forall \mathbf{a} \in \mathbf{L}: \mathbf{a} \oplus \mathbf{0} = \mathbf{a}$ ($\mathbf{0}$ = neutrální prvek vzhledem k sečítání; určen jednoznačně)

$\forall \mathbf{a} \in \mathbf{L} \exists -\mathbf{a} \in \mathbf{L}: \mathbf{a} \oplus (-\mathbf{a}) = \mathbf{0}$ (opačné prvky; jsou určeny jednoznačně)

$\forall \mathbf{a} \in \mathbf{L} \forall t \in T: \exists t \otimes \mathbf{a} \in \mathbf{L}$ (násobení vektoru skalárem, součin je určen jednoznačně)

$\forall \mathbf{a}, \mathbf{b} \in \mathbf{L} \forall s, t \in T: t \otimes (\mathbf{a} \oplus \mathbf{b}) = t \otimes \mathbf{a} \oplus t \otimes \mathbf{b}$

$\forall \mathbf{a} \in \mathbf{L} \forall s, t \in T: (s \cdot t) \otimes \mathbf{a} = s \otimes (t \otimes \mathbf{a})$

$\forall \mathbf{a} \in \mathbf{L} \forall s, t \in T: (s + t) \otimes \mathbf{a} = s \otimes \mathbf{a} \oplus t \otimes \mathbf{a}$

$\forall \mathbf{a} \in \mathbf{L} : 1 \otimes \mathbf{a} = \mathbf{a}$, kde 1 je jednotkový prvek tělesa T

$\forall \mathbf{a} \in \mathbf{L} : 0 \otimes \mathbf{a} = \mathbf{0}$, kde 0 je nulový prvek tělesa T

V případě součtu dvou n -tic vznikne výsledná n -tice součty „bit po bitu“, čili

$\mathbf{c} = \mathbf{a} \oplus \mathbf{b} \Leftrightarrow \forall i = 1, 2, \dots, n: c_i = a_i + b_i$, kde $+$ je operace sečítání nad tělesem T .

Připomenutí z lineární algebry

V případě binárních kódů vytvoříme těleso zavedením těchto operací:

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| · | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Opačným prvkem k 0 je tedy 0, opačným prvkem k 1 je 1, inverzním prvkem k 1 je 1.

Množina $\{0,1\}$ s takto definovanými operacemi sečítání a násobení bývá označována jako *těleso* Z_2 .

Parita a opakovací kód jako lineární kódy

Kód celkové kontroly parity s délkou značky $n = 5$ (k binárnímu slovu přidáme jeden kontrolní znak tak, aby celkový počet jedniček byl sudý).

Protože je ve značce sudý počet jedniček, lze je „posčítat po dvou“.

V tělese Z_2 platí $1 + 1 = 0$.

Každá kódová značka $\mathbf{v} \in \mathbf{K}$ tedy splňuje podmínku $v_1 + v_2 + v_3 + v_4 + v_5 = 0$

Opakovací kód délky $n = 5$ (přenášený znak kódové abecedy n -krát zopakujeme).

Všechny znaky ve značce jsou opakováním znaku v_1

Každá kódová značka $\mathbf{v} \in \mathbf{K}$ splňuje podmínky $v_1 = v_2$, $v_1 = v_3$, $v_1 = v_4$, $v_1 = v_5$

tedy $v_1 + v_2 = 0$, $v_1 + v_3 = 0$, $v_1 + v_4 = 0$, $v_1 + v_5 = 0$

Parita a opakovací kód jako lineární kódy

- Zobecnění
 - oba kódy jsme popsali soustavou homogenních lineárních rovnic (paritní kód délky n jednou rovnicí, opakovací kód délky n soustavou $n - 1$ rovnic)
 - tyto rovnice se nazývají *kontrolní rovnice*
 - počet kontrolních rovnic je dán počtem *kontrolních znaků*, které jsme přidali k *informačním znakům*
 - kódové značky obou kódů jsou řešením (soustavy) kontrolních rovnic
- **Řešení soustavy homogenních lineárních rovnic o n proměnných tvoří lineární prostor, jenž je podprostorem lineárního prostoru T^n .**

Binární lineární kód - definice

- Binární kód K je lineárním kódem, jestliže je podprostorem lineárního prostoru Z_2^n
- Dimenzi k lineárního kódu nazýváme *počtem informačních znaků*.
- Pro lineární kódy s k informačními znaky používáme název *lineární (n, k) kód*.
- Součet libovolných dvou značek lineárního kódu je také kódová značka.
- Prvkem každého lineárního kódu je nulová značka.
- **Pozor!** Tvrzení „Kód má k informačních znaků“ obecně nemusí znamenat, že ve značkách dokážeme rozlišit, které znaky jsou informační a které kontrolní.
Pojem „počet informačních znaků“ vyjadřuje něco obecnějšího.

Kód s k informačními znaky - definice

Blokový kód $K \subseteq T^n$ délky n má k *informačních znaků* (a $n - k$ kontrolních znaků) právě tehdy, jestliže existuje prosté zobrazení φ množiny všech slov délky k na množinu kódových značek K , tedy $\varphi: T^k \rightarrow K$.

Používaná terminologie a značení:

$\varphi: T^k \rightarrow K$ nazýváme *kódování informačních znaků*

$u \in T^k$ nazýváme *informační část*

$v \in K$, $v = \varphi(u)$ je *kódová značka*

(n, k) kód je kód s k informačními a $n - k$ kontrolními znaky

POZOR! Termín dekódování je používán ve dvou významech:

$\delta: T^n \rightarrow K$ umožňuje opravy t – násobných chyb

$\varphi^{-1}: K \rightarrow T^k$ z kódové značky „extrahuje“ informační část

Kód s k informačními znaky - dekodování

Zpracování přijaté n – tice (v ideálním případě, kdy neexistují „neopravitelné“ n – tice):

$$u' = \varphi^{-1}(\delta(w)) \quad , \text{ kde}$$

$w \in T^n$ je přijatá n – tice (ovlivněná šumem)

$\delta(w) \in K$ je přijatá n – tice po opravě chyby (tedy kódová značka)

$u' \in T^k$ je informační část extrahovaná z přijaté n – tice po opravě chyby

Připomenutí z lineární algebry

- Každý lineární prostor je jednoznačně určen svojí bází.
- Každý prvek lineárního prostoru je v dané bázi jednoznačně určen svými souřadnicemi.
- Bázi lineárního kódu bude tvořit k lineárně nezávislých značek.
- Každá kódová značka bude jednoznačně určena svými souřadnicemi, tedy k – prvkovým vektorem, který představuje informační znaky, jež jsou ve značce zakódovány.
- Kódování informační části pak má tvar lineární kombinace bázových značek:
$$\mathbf{v} = \varphi(u) = u_1 \cdot \mathbf{b}_1 + u_2 \cdot \mathbf{b}_2 + \cdots + u_k \cdot \mathbf{b}_k$$
 kde u je vektor informačních znaků

Lineární kódy – používané značení

\mathbf{u} typu $k/1$ (sloupeček o k prvcích) je informační část (vektor informačních znaků)

\mathbf{v} typu $n/1$ (sloupeček o n prvcích) je kódová značka (tj. zakódovaná informační část)

\mathbf{e} typu $n/1$ (sloupeček o n prvcích) je chybový vektor

\mathbf{w} typu $n/1$ (sloupeček o n prvcích) je přijatá n – tice ($\mathbf{w} = \mathbf{v} + \mathbf{e}$)

\mathbf{s} typu $(n - k)/1$ (sloupeček o $n - k$ prvcích) je *syndrom* (výsledek kontroly přijaté n – tice)

Vektory budeme obvykle zapisovat ve formě transponovaných řádků, např. $\mathbf{u} = [001110]^T$ nebo $\mathbf{v} = [v_1 v_2 \dots \dots v_n]^T$.

Lineární kódy – používané značení

- Každá kódová značka lineárního kódu musí být řešením soustavy kontrolních rovnic.
- Matici této soustavy ve tvaru s nulovou pravou stranou nazýváme *kontrolní matice*. Značíme-ji \mathbf{H} , je typu $(n - k) / n$. Pro $n - k$ se používá značení r .
- Řádky kontrolní matice jsou bazové prvky ortogonálního doplňku ke kódu K .
- Vlastnosti kontrolní matice:
 - řádky kontrolní matice jsou lineárně nezávislé
 - slovo w je kódovou značkou právě tehdy, když $\mathbf{s} = \mathbf{H} \cdot \mathbf{w} = \mathbf{0}$.

Lineární kódy – používané značení

- Uspořádáme-li prvky báze lineárního kódu do matice tak, že každý bázeový prvek bude řádkem této matice, vytvoříme tak *generující matici*. Značíme-ji \mathbf{G} , je typu k/n .

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_k \end{bmatrix} = \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{k1} & \vdots & b_{kn} \end{bmatrix}$$

- Pak lze kódování informační části $\mathbf{v} = u_1 \cdot \mathbf{b}_1 + u_2 \cdot \mathbf{b}_2 + \dots + u_k \cdot \mathbf{b}_k$ vyjádřit jako $\mathbf{v} = \mathbf{G}^T \cdot \mathbf{u}$.

Lineární kódy – používané značení

- Vlastnosti generující matice:
 - každý řádek generující matice je kódovou značkou
 - řádky generující matice jsou lineárně nezávislé
 - každé kódové slovo je lineární kombinací řádků generující matice a je jednoznačně určeno informačními znaky
 - skalární součin libovolného řádku generující matice s libovolným řádkem kontrolní matice je nulový
- Ke generující matici \mathbf{G} vždy existuje kontrolní matice \mathbf{H} s vlastností $\mathbf{H} \cdot \mathbf{G}^T = \mathbf{0}$, není ale určena jednoznačně, může existovat více různých kontrolních matic.

Ilustrační příklad

- Kód celkové parity s délkou značky $n = 5$ (tedy $k = 4$).
- Jak vytvořit bázi kódu? Zakódováním kanonické báze lineárního prostoru všech informačních částí:

$$\mathbf{b}_1 = [1000\mathbf{1}], \quad \mathbf{b}_2 = [0100\mathbf{1}], \quad \mathbf{b}_3 = [0010\mathbf{1}], \quad \mathbf{b}_4 = [0001\mathbf{1}]$$

- Ukázka zakódování $\mathbf{u} = [1010]$ prostou lineární kombinací bázevých prvků:

$$\begin{aligned} \mathbf{v} &= u_1 \cdot \mathbf{b}_1 + u_2 \cdot \mathbf{b}_2 + u_3 \cdot \mathbf{b}_3 + u_4 \cdot \mathbf{b}_4 = \\ &= 1 \cdot [1000\mathbf{1}] + 0 \cdot [0100\mathbf{1}] + 1 \cdot [0010\mathbf{1}] + 0 \cdot [0001\mathbf{1}] = [1010\mathbf{0}] \end{aligned}$$

Ilustrační příklad

- Uspořádání báзовých prvků do generující matice:

$$G = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} \end{bmatrix} = \begin{bmatrix} 1000\mathbf{1} \\ 0100\mathbf{1} \\ 0010\mathbf{1} \\ 0001\mathbf{1} \end{bmatrix}$$

- Ukázka zakódování $u = [1010]$ maticovým násobením:

$$v = G^T \cdot u = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \\ \mathbf{1111} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \mathbf{0} \end{bmatrix}$$

Praktický postup:

$$\begin{bmatrix} 1000\mathbf{1} \\ 0100\mathbf{1} \\ 0010\mathbf{1} \\ 0001\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ 0 \\ \mathbf{1} \\ 0 \end{bmatrix}$$

G

u

Vybrané
řádky

1000 $\mathbf{1}$

0010 $\mathbf{1}$

$\overline{[1010\mathbf{0}]}$ v

Ilustrační příklad

- Kontrolní matice H je maticí soustavy kontrolních rovnic (u paritního kódu jednou kontrolní rovnicí) $v_1 + v_2 + v_3 + v_4 + v_5 = 0$.
- Kontrolní matice je tedy $H = [11111]$.
- Kontrola přijaté značky s chybou ve třetím přenášeném znaku:

$$v = [10100]^T \quad e = [00100]^T \quad w = v + e = [10100]^T + [00100]^T = [10000]^T$$

$$s = H \cdot w = [11111] \cdot [10000]^T = [1]$$

Praktický postup:
$$\begin{array}{rcl} w & [10000]^T \\ H & [11111] \\ \text{Vybrané sloupce} & [1 \quad] \quad | \quad [1] \quad s \end{array}$$

Ilustrační příklad

- Opakovací kód s délkou značky $n = 5$ (tedy $k = 1, r = n - k = 4$).
- Jak vytvořit bázi kódu? Zakódováním kanonické báze lineárního prostoru

všech informačních částí: $\mathbf{b}_1 = [11111]^T$

- Generující matice: $\mathbf{G} = [11111]$

- Do množiny kódových značek patří pouze dvě značky:

$$0 \cdot \mathbf{b}_1 = 0 \cdot [11111]^T = [00000]^T \quad 1 \cdot \mathbf{b}_1 = 1 \cdot [11111]^T = [11111]^T$$

- Připomenutí kontrolních rovnic:

$$v_1 + v_2 = 0, v_1 + v_3 = 0, v_1 + v_4 = 0, v_1 + v_5 = 0$$

Ilustrační příklad

- Opakovací kód s délkou značky $n = 5$ (tedy $k = 1, r = n - k = 4$).

- Připomenutí kontrolních rovnic; kontrolní matice :

$$v_1 + v_2 = 0, v_1 + v_3 = 0, v_1 + v_4 = 0, v_1 + v_5 = 0$$

$$H = \begin{bmatrix} 11000 \\ 10100 \\ 10010 \\ 10001 \end{bmatrix}$$

- Kontrola přijaté značky při vyslání $\mathbf{v} = [1\mathbf{1111}]^T$ a chybě ve 3. a 4. bitu.

$$\mathbf{e} = [00110]^T \quad \mathbf{w} = \mathbf{v} + \mathbf{e} = [11111]^T + [00110]^T = [11001]^T$$

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{w} = \begin{bmatrix} 11000 \\ 10100 \\ 10010 \\ 10001 \end{bmatrix} \cdot [11001]^T = [0110]^T$$

$$\begin{matrix} & & & & [\mathbf{11001}] \\ & & & & \begin{bmatrix} 11000 \\ 10100 \\ 10010 \\ 10001 \end{bmatrix} \end{matrix} \quad \mathbf{s} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Podrobnější pohled na generující matici

Binární lineární kód je zadán generující maticí

$$G_1 = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Kód generovaný maticí G_1 je lineárním obalem řádků generující matice:

$$\mathbf{K} = \{ \mathbf{v} | \mathbf{v} = u_1 \cdot \mathbf{b}_1 + u_2 \cdot \mathbf{b}_2 + u_3 \cdot \mathbf{b}_3 + u_4 \cdot \mathbf{b}_4, \quad \forall u_1, u_2, u_3, u_4 \in \mathbf{Z}_2 \}$$

- Podle sloupců matice lze napsat „vytvorující rovnice“ pro prvky značky:

$$\begin{aligned} v_1 &= u_1 & , & & v_2 &= \mathbf{u}_2 & , & & v_3 &= u_3 & , & & v_4 &= u_4 \\ v_5 &= \mathbf{u}_2 + \mathbf{u}_3 + \mathbf{u}_4 & , & & v_6 &= \mathbf{u}_1 + \mathbf{u}_3 + \mathbf{u}_4 & , & & v_7 &= \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_4 \end{aligned}$$

Podrobnější pohled na generující matici

Uvažujme jiný lineární (7,4) kód generovaný maticí G_2 :

$$G_2 = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}$$

- Podle sloupců matice lze napsat „vytvorující rovnice“ pro prvky značky:

$$\begin{aligned} v_1 &= u_1 & , & & v_2 &= u_1 + u_2 & , & & v_3 &= u_2 + u_3 & , & & v_4 &= u_1 + u_3 + u_4 \\ v_5 &= u_2 + u_4 & , & & v_6 &= u_3 & , & & v_7 &= u_4 \end{aligned}$$

- V prvcích v_1, v_6, v_7 jsou přenášeny informační znaky „napřímo“.
- Informační znak u_2 není ve značce obsažen „v čisté podobě“.
- Přesto říkáme, že kód má 4 informační prvky (existuje zobrazení $\varphi: \mathbf{T}^4 \rightarrow \mathbf{K}$).

Systematické kódy

- *Systematický kód* – kód, v němž jsou informační znaky umístěny na začátku kódové značky při zachování jejich pořadí, platí tedy

$$v_1 = u_1, \quad v_2 = u_2, \quad \dots, \quad v_k = u_k$$

- Generující matice systematického kódu má tvar

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1(n-k)} \\ 0 & 1 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2(n-k)} \\ & & & & \dots & \dots & \dots & \\ & & & & \dots & \dots & \dots & \\ 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{k(n-k)} \end{bmatrix} = [\mathbf{I}_k \mid \mathbf{B}]$$

Systematické kódy

$$\mathbf{G} = \begin{bmatrix} 10 \dots 0 & b_{11}b_{12} \dots b_{1(n-k)} \\ 01 \dots 0 & b_{21}b_{22} \dots b_{2(n-k)} \\ & \dots \dots \dots \\ & \dots \dots \dots \\ 00 \dots 1 & b_{k1}b_{k2} \dots b_{k(n-k)} \end{bmatrix} = [\mathbf{I}_k \mid \mathbf{B}]$$

- Pro $i > k$ platí $v_i = u_1 \cdot b_{1i} + u_2 \cdot b_{2i} + \dots + u_k \cdot b_{ki}$.
- K systematickému kódu s generující maticí ve tvaru $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{B}]$ je jednou z kontrolních matic matice ve tvaru $\mathbf{H} = [-\mathbf{B}^T \mid \mathbf{I}_{n-k}]$
- Pro binární kódy je znaménko – u matice \mathbf{B}^T nadbytečné ($-1 = 1$).

Minimální vzdálenost v lineárním kódu

- Minimální Hammingova vzdálenost kódu K : $d_0(K) = \min_{u,v \in K, u \neq v} d(u,v)$

- V případě, že K je lineární kód:

Předpokládejme, že u a v jsou dvě kódové značky, které určují $d_0(K)$.

Kódovými značkami jsou i „opačná“ značka $-u$

a „součtová značka“ $v + (-u) = v - u$.

Platí $d_0 = d(u,v) = d(u - u, v - u) = d(0, v - u) = \|v - u\|$

$\|v - u\|$ představuje počet nenulových prvků ve značce $v - u$, tj. *Hammingovu váhu značky $v - u$*

- Minimální Hammingova vzdálenost v lineárním kódu je rovna minimální Hammingově váze nenulové značky.