Kapitola 1

Kódovanie

1.1 Prenosový reťazec

Všeobecná schéma prenosového reťazca je nasledujúca:

Zdroj signálu → Kóder → Prenosový kanál → Dekóder → Príjímač

Zdroj signálu, prenosový kanál a prijímacie zariadenie môžu pracovať v úplne iných abecedách. Vysielacie štúdio má hudobnú skladbu uloženú na CD nosiči – teda v binárnom kóde. Aby ju mohlo vysielať FM prenosom, musí z tohoto binárneho kódu vytvoriť vysokofrekvenčný signál (okolo 100 MHz, čo je signál, ktorým pracuje prenosový kanál) čo nie je nič iného, ako zakódovanie. Rádiový prijímač prijme FM signál a spracuje ho do konečnej podoby zvukových vĺn - dekóduje signál kanálu.

Ak chceme preniesť správu pomocou baterky schopnej vydávať len signály bodka, čiarka a medzipísmenová medzera, musíme prenášaný text upraviť - zakódovať pomocou napr. Morseovej abecedy do týchto znakov.

Hlavným dôvodom kódovania správ je prispôsobenie abecedy zdroja abecede prenosového kanála. Pritom však môžeme mať i ďalšie ciele – napríklad chceme, aby zakódovaná správa bola čo najkratšia (kompresia správ). Na druhej strane môžeme žiadať, aby bolo možné poznať, či v zakódovanej správe nedošlo behom prenosu k jednej alebo viacerým chybám, resp. aby zakódovaná správa bola odolná voči jednej či viacerým chybám pri prenose. Naviac chceme, aby kompresia, zisťovanie chyby, či odstraňovanie chyby boli výpočtovo čo najjednoduchšie. Jednotlivé požiadavky sú protichodné a nie je vždy jednoduché zaistiť ich samotné, nie to ešte v kombináciách. Kódovanie nemá za cieľ utajenie správy, preto je nutné rozlišovať ho od šifrovania – kryptografie, ktorých cieľom ochrana a utajenie dát. Kryptografie sa dotkneme v poslednej kapitole tejto publikácie.

Problémami kódovania, kompresie dát, kódov odhaľujúcich chyby a samoopravných kódov sa zaoberá teória kódovania, ktorej základy uvedieme v tejto kapitole.

1.2 Abeceda, kód a kódovanie

Nech $A = \{a_1, a_2, \ldots, a_r\}$ je konečná r-prvková množina. Prvky množiny A nazveme **znakmi**, množinu A **abecedou**. Množinu

$$A^* = \bigcup_{i=1}^{\infty} A^i \tag{1.1}$$

nazveme **množinou všetkých slov abecedy** A. **Dĺžka slova** $\mathbf{a} \in A^*$ je počet znakov slova $\mathbf{a}$.
Na množine slov abecedy A zavádzame binárnu operáciu zreťazenia slov: ak $\mathbf{b} = b_1 b_2 \ldots b_p$, $\mathbf{c} = c_1 c_2 \ldots c_q$ sú dve slová z A^*, potom definujeme

$$\mathbf{b}|\mathbf{c} = b_1 b_2 \ldots b_p c_1 c_2 \ldots c_q \tag{1.2}$$

Zreťazenia slov píšeme bez medzery, či iného oddeľovacieho znaku. Každé slovo môžeme považovať za zreťazenie jeho častí ľubovoľným spôsobom, ako sa nám hodí. Tak napríklad $01010001 = 0101|0001 = 010|100|01 = 0|1|0|1|0|0|0|1$.

Nech $A = \{a_1, a_2, \ldots, a_r\}$, $B = \{b_1, b_2, \ldots, b_s\}$ sú dve abecedy. **Kódovanie** je zobrazenie

$$K : A \to B^*, \tag{1.3}$$

t.j. predpis, ktorý každému prvku abecedy A priradí slovo abecedy B. Abeceda A je **zdrojová abeceda**, jej znaky sú **zdrojové znaky**, abeceda B je **kódová abeceda** a jej znaky sú **kódové znaky**. Množinu všetkých slov kódovej abecedy typu

$$\mathcal{K} = \{\mathbf{b} \mid \mathbf{b} = K(a),\ a \in A\} = \{K(a_1), K(a_2), \ldots, K(a_r)\} \tag{1.4}$$

nazveme **kódom**, každé slovo z množiny $\mathcal{K}$ je **kódové slovo** ostatné slová z abecxedy B sú **nekódové slová**. Význam majú iba prosté kódovania, t.j. také, kde rôznym zdrojovým znakom $a_i,\ a_j \in A$ zodpovedajú rôzne kódové slová $K(a_i),\ K(a_j)$, preto budeme vždy predpokladať, že zobrazenie K je prosté.

Každé kódovanie K môžeme rozšíriť na kódovanie K^* zdrojových slov predpisom

$$K^*(a_{i_1}a_{i_2}\ldots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_n}) \tag{1.5}$$

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priadiť kódové slová rôznej dĺžky. často sa však stretávame s kódavaniami u ktorých všetky kódové slová majú rovnakú dĺžku. **Blokové kódovanie** (dĺžky n) je také kódovanie, ktoré všetkým zdrojovým znakom priradí kódové slová rovnakej dĺžky n.

Príklad 1.1. Nech $A = \{a, b, c, d\}$, $B = \{0, 1\}$, nech $K(a) = 00,\ K(b) = 01,\ K(c) = 10,\ K(d) = 11$. Potom správu $aabd$ (t.j. slovo v abecede A) zakódujeme ako $K^*(aabd) = 00000111$. Ak na strane prijímača dostaneme slovo 00000111 a poznáme zobrazenie K, vieme, že každý znak zdrojovej abecedy bol zakódovaný do dvoch znakov kódovej abecedy a teda jediné možné rozdelenie prijatej správy na kódové slová je 00|00|01|11, čo vedie k jednoznačnému dekódovaniu správy. Kódovanie K je blokovým kódovaním dĺžky 2.

Príklad 1.2. Študenti sú hodnotení známkami 1, 2, 3, 4. Vieme, že najčastejšia známka je 2 a potom 1. Na zakódovanie štyroch znakov zdrojovej abecedy $A = \{1, 2, 3, 4\}$ by stačili dva znaky binárnej kódovej abecedy $B = \{0, 1\}$. Pretože však trojky a štvorky sa vyskytujú zriedkavo, a dvojky zas veľmi často, chceme dvojkám dať čo najkratšie kódové slovo. Navrhneme preto toto kódovanie: $K(1) = 01,\ K(2) = 0,\ K(3) = 011,\ K(4) = 111$. Správa 1234 bude zakódovaná ako 01|0|011|111. Ak budeme postavení pred úlohu dekódovať správu 010011111, budeme musieť postupovať od zadu. Ak napríklad dostaneme čiastočnú správu $01111\ldots$, nevieme, či bola vyslaná ako $0|111|1\ldots$, alebo $01|111\ldots$, alebo $011|11\ldots$, nemôžeme ho preto dekódovať znak po znaku.

Definícia 1.1. Hovoríme, že kódovanie $K : A \to B^*$ je **jednoznačne dekódovateľné**, ak zo znalosti zakódovanej správy $K^*(a_1a_1\ldots a_n)$ môžeme vždy určiť zdrojovú správu $a_1a_1\ldots a_n$, t.j. ak je zobrazenie $K^* : A^* \to B^*$ prostým zobrazením.

Príklad 1.3. Rozšírme zdrojovú abecedu z príkladu 1.2 na $A = \{1, 2, 3, 4, 5\}$ a definujme kódovnie $K(1) = 01,\ K(2) = 0,\ K(3) = 011,\ K(4) = 111,\ K(5) = 101$. Majme správu 0101101. Pre dekódovanie by sme ju mohli rozdeliť nasledovne: 0|101|101, 01|01|101, 01|011|01, pričom tieto delenia zodpovedajú zdrojovým slovám porade 255, 115, 131. Vidíme, že napriek tomu, že kódové zobrazenie $K : A \to B^*$ je prosté, príslušné zobrazenie $K^* : A^* \to B^*$ prosté nie je. K nie je jednoznačne dekódovateľné kódovanie.

1.3 Prefixové kódovanie a Kraftova nerovnosť

Definícia 1.2. Prefixom slova $\mathbf{b} = b_1 b_2 \dots b_k$ nazveme každé zo slov b_1, $b_1 b_2$, $\dots$, $b_1 b_2 \dots b_{k-1}$, $b_1 b_2 \dots b_k$. Kódovanie resp. kód sa nazýva **prefixové**, ak žiadne kódové slovo nie je prefixom iného kódového slova.

Všimnime si, že každé blokové kódovanie je prefixovým kódovaním. Rozšíreným prefixovým kódovaním, s ktorým sa všetci dennodenne stretávame je priradenie telefónnych čísel staniciam v sieti Slovenských telekomunikácií. Telefónne čísla nie sú blokovým kódom, pretože čísla staníc majú rôznu dĺžku – napr, 120 – informácie resp. 155 – záchranná služba sú len trojmiestne, ale pre všetky stanice s troma dekadickými číslicami nevystačíme. Číslo žiadnej telefónnej stanice nemôže začínať číslom inej stanice, pretože by sa v priebehu vytáčania dlhšieho čísla vždy ohlásila stanica s prefixom. Tak napríklad číslo 120 je vyhradené pre informácie a žiadne iné telefónne číslo v sieti Slovenských telekomunikácií nemá číslo typu 120XXX.

Prefixové kódovanie je jediné kódovanie, ktoré môžeme dekódovať znak po znaku – t.j. v priebehu prijímania správy (a nemusíme čakať na prijatie celej správy). Dekódovanie prijatej správy robíme tak, že v nej nájdeme najmenší počet znakov zľava, ktoré tvoria kódové slovo $K(a)$ niektorého zdrojového znaku a, tieto znaky dekódujeme, zrušíme dekódované znaky z kódovanej správy a pokračujeme ďalej rovnakým spôsobom.

Veta 1.1. Kraftova nerovnosť. *Majme zdrojovú abecedu $A = \{a_1, a_2, \dots, a_r\}$ s r znakmi, kódovú abecedu $B = \{b_1, b_2, \dots, b_n\}$ s n znakmi. Prefixový kód s dĺžkami kódových slov $d_1, d_2, \dots, d_r$ existuje práve vtedy, keď*

$$n^{-d_1} + n^{-d_2} + \dots + n^{-d_r} \leq 1. \tag{1.6}$$

Dôkaz. Nech platí Kraftova nerovnosť (1.6). Usporiadajme znaky zdrojovej abecedy tak, aby platilo $d_1 \leq d_2 \leq \dots \leq d_r$. Za $K(a_1)$ zvoľme ľubovoľné slovo abecedy B dĺžky d_1. Predpokladajme, že už máme priradené kódové slová požadovanej dĺžky $K(a_1)$, $K(a_2)$, $\dots$, $K(a_i)$. Pri voľbe kódového slova $K(a_{i+1})$ dĺžky d_{i+1} sa musíme vyhnúť $n^{(d_{i+1}-d_1)}$ slovám dĺžky d_{i+1}, ktoré majú prefix $K(a_i)$, $n^{(d_{i+1}-d_2)}$ slovám dĺžky d_{i+1}, ktoré majú prefix $K(a_{i+1})$ atď. až $n^{(d_{i+1}-d_i)}$ slovám dĺžky d_{i+1}, ktoré majú prefix $K(a_i)$, pričom všetkých slov dĺžky d_{i+1} je $n^{d_{i+1}}$. Počet zakázaných slov je teda

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \dots + n^{(d_{i+1}-d_i)}. \tag{1.7}$$

Keďže platí (1.6), tým skôr platí pre prvých $i+1$ členov ľavej strany (1.6):

$$n^{-d_1} + n^{-d_2} + \dots + n^{-d_i} + n^{-d_{i+1}} \leq 1. \tag{1.8}$$

Po vynásobení nerovnosti (1.8) číslom $n^{d_{i+1}}$ dostávame

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \dots + n^{(d_{i+1}-d_i)} + 1 \leq n^{d_{i+1}}. \tag{1.9}$$

Podľa (1.9) je počet zakázaných slov aspoň o 1 slovo menší, než počet všetkých slov dĺžky d_{i+1} a preto môžeme toto slovo definovať ako kódové slovo $K(a_{i+1})$.

Majme prefixový kód s dĺžkami $d_1, d_2, \dots, d_r$. Predpokladajme $d_1 \leq d_2 \leq \dots \leq d_r$. Existuje n^{d_r} slov dĺžky d_r, ktorými možno zakódovať písmeno a_r. Pre každé $i = 1, 2, \dots, r-1$ je slovo $K(a_i)$ prefixom $n^{(d_r-d_i)}$ slov dĺžky d_r – tieto slová sú pre výber slova $K(a_r)$ zakázané (inak by totiž kód nebol prefixový). Pretože aj pre slovo $K(a_r)$ sa ušlo jedno kódové slovo dĺžky d_r, musí platiť:

$$n^{(d_r-d_1)} + n^{(d_r-d_2)} + \dots + n^{(d_r-d_{r-1})} + 1 \leq n^{d_r} \tag{1.10}$$

Vydelením nerovnosti (1.10) číslom n^{d_r} dostávame požadovanú Kraftovu nerovnosť (1.6). □

Poznámka 1.1. Algoritmus na zostrojenie prefixového kódu s dĺžkami slov $d_1, d_2, \ldots, d_r$**.** Prvá časť dôkazu vety 1.1 je konštruktívna, dáva návod na zostrojenie prefixového kódovania, ak sú dané požadované dĺžky $d_1 \leq d_2 \leq \cdots \leq d_r$ kódových slov splňujúce Kraftovu nerovnosť. Za $K(a_1)$ zvolíme ľubovoľné slovo dĺžky d_1. Keď už máme určené $K(a_1),\ K(a_2),\ \ldots,\ K(a_i)$, za $K(a_{i+1})$ zvolíme ľubovoľné slovo dĺžky d_{i+1}, ktoré nemá prefix $K(a_1)|K(a_2)|\ldots,\ K(a_i)$. Existenciu aspoň jedného takéhoto slova zaručuje Kraftova nerovnosť.

Veta 1.2. Mac Millan. *Pre každé jednoznačne dekódovateľné kódovanie so zdrojovou abecedou* $A = \{a_1, a_2, \ldots, a_r\}$ *a kódovou abecedou* $B = \{b_1, b_2, \ldots, b_n\}$ *s dĺžkami kódových slov* $d_1, d_2, \ldots, d_r$ *platí Kraftova nerovnosť* (1.6).

Dôkaz. Majme jednoznačne dekódovateľné kódovanie K s dĺžkami kódových slov $d_1 \leq d_2 \leq \cdots \leq d_r$. Označme

$$c = n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r} \tag{1.11}$$

V ďalšom postupe sa budeme snažiť ukázať, že $c \leq 1$.
Nech k je ľubovoľné prirodzené číslo. Majme množinu $\mathcal{M}_k$ všetkých slov kódovej abecedy typu $\mathbf{b} = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k})$. Dĺžka každého takéhoto slova $\mathbf{b}$ je $d_{i_1} + d_{i_2} + \cdots + d_{i_k}$ a je menšia alebo rovná $k.d_r$, pretože maximálna dĺžka kódového slova je d_r.
Skúmajme výraz

$$c^k = \left[n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r}\right]^k = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_k=1}^{n} n^{-(d_{i_1}+d_{i_2}+\cdots+d_{i_k})} \tag{1.12}$$

Pretože K je jednoznačne dekódovateľné, platí pre dve rôzne slová zdrojovej abecedy $a_1, a_2, \ldots, a_n$, $a'_1, a'_2, \ldots, a'_n$ $K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k}) \neq K(a'_{i_1})|K(a'_{i_2})|\ldots|K(a'_{i_k})$. Preto ku každému slovu $\mathbf{b} = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k})$ z množiny $\mathcal{M}_k$ možno priradiť práve jeden sčítanec $n^{-(d_{i_1}+d_{i_2}+\cdots+d_{i_k})}$ na pravej strane (1.12) taký, že jeho záporne vzatý exponent $(d_{i_1} + d_{i_2} + \cdots + d_{i_k})$ sa rovná dĺžke slova $\mathbf{b}$. Ako sme už ukázali, maximálna dĺžka slova z množiny $\mathcal{M}_k$ je $k.d_r$. Označme $M = k.d_r$ Výraz na pravej strane vzťahu (1.12) je polynomom stupňa M premennej $\dfrac{1}{n}$, a preto ho môžeme zapísať v tvare

$$c^k = s_1.n^{-1} + s_2.n^{-2} + \cdots + s_M.n^{-M} = \sum_{i=1}^{M} s_i.n^{-i} \tag{1.13}$$

V súčte na pravej strane (1.12) sa vyskytuje člen n^{-i} práve toľkokrát, koľko slov z množiny $\mathcal{M}_k$ má dĺžku i. Pretože kódová abeceda má n znakov, najviac n^i slov z množiny $\mathcal{M}_k$ môže mať dĺžku i, čo znamená, že $s_i \leq n^i$. Môžeme teda písať:

$$\begin{aligned} c^k = s_1.n^{-1} + s_2.n^{-2} + \cdots + s_M.n^{-M} \leq \\ \leq n^1.n^{-1} + n^2.n^{-2} + \cdots + n^M.n^{-M} \leq= 1 + 1 + \cdots + 1 = M = k.d_r \end{aligned} \tag{1.14}$$

a teda

$$\frac{c^k}{k} \leq d_r \tag{1.15}$$

Pretože nerovnosť (1.15) musí platiť pre ľubovoľné k, musí byť $c \leq 1$. □

Dôsledkom Mac Millanovej vety je, že žiadnym jednoznačne dekódovateľným kódovaním nedosiahneme kratšie dĺžky kódových slov ako prefixovým kódovaním. Keďže prefixové kódovanie má veľa výhod – jednoduché dekódovanie znak po znaku, netreba čakať na príjem celej správy – stačí sa obmedziť na prefixové kódovanie.

1.4 Najkratší kód - Huffmanova konštrukcia

Nech je daný stacionárny zdroj $\mathcal{Z} = (\Omega, \mathcal{A}, P)$, ktorý produkuje jednotlivé znaky zdrojovej abecedy $A = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami $p_1, p_2, \ldots, p_r$, $\sum_{i=1}^{r} p_i = 1$. Majme prefixové kódovanie K také, že dĺžky kódových slov $K(a_1),\ K(a_2),\ \ldots,\ K(a_r)$ sú $d_1, d_2, \ldots, d_r$. Potom **stredná dĺžka kódového slova** kódovania K je

$$d(K) = p_1.d_1 + p_2.d_2 + \cdots + p_r.d_r = \sum_{i=1}^{r} p_i.d_i \tag{1.16}$$

Ak kódovaním K kódujeme správu s veľkým počtom N znakov, môžeme očakávať, že dĺžka (počet znakov) zakódovanej správy v abecede B bude približne $N.d(K)$. Keďže veľmi často (z hľadiska prenosu alebo uloženia správy) chceme, aby zakódovaná správa bola čo najkratšia, hľadáme kódovanie K s minimálnym $d(K)$.

Definícia 1.3. Majme danú zdrojovú abecedu $A = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami výskytu $p_1, p_2, \ldots, p_r$ a kódovú abecedu $B = \{b_1, b_2, \ldots, b_n\}$. **Najkratšie n-znakové kódovanie** abecedy A je také kódovanie $K : A \to B^*$, ktoré má najmenšiu strednú dĺžku kódového slova $d(K)$.

Najkratší prefixový kód skonštruoval O. Huffman (čítaj hafmen) v roku 1952. Budeme sa zaoberať hlavne binárnym kódovaním, pretože je z hľadiska aplikácií najdôležitejšie. Predpokladáme, že zdrojové znaky $a_1, a_2, \ldots, a_r$ sú usporiadné zostupne podľa pravdepodobnosti, t.j. $p_1 \geq p_2 \geq \cdots \geq p_r$.

Ak máme len dva znaky $a_1,\ a_2$ s ľubovoľnými pravdepodobnosťami, je sitácia jednoduchá – najkratšie kódovanie je $K(a_1) = 0$, $K(a_2) = 1$ (alebo $K(a_1) = 1$, $K(a_2) = 0$).

Definícia 1.4. Majme abecedu s r znakmi $A = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami znakov $p_1 \geq p_2 \geq \cdots \geq p_r$. **Redukovanou abecedou** abecedy A nazveme abecedu $\tilde{A} = \{\tilde{a}_1, \tilde{a}_2, \ldots, \tilde{a}_{r-1}\}$, kde $\tilde{a}_i = a_i$ pre $i = 1, 2 \ldots, r-2$, $\tilde{a}_{r-1} = a^*$, kde $a^* \neq A$ a $p(\tilde{a}_i) = p(a_i)$ pre $i = 1, 2 \ldots, r-2$, $p(\tilde{a}_{r-1}) = p_{r-1} + p_{r-2}$.

Veta 1.3. *Nech $\tilde{A}$ je redukovná abeceda abecedy A, nech $\tilde{K}$ je najkratšie binárne kódovanie redukovanej abecedy $\tilde{A}$. Potom kódovanie K definované*

$$K(a_1) = \tilde{K}(\tilde{a}_1), \quad K(a_2) = \tilde{K}(\tilde{a}_2), \quad \ldots, \quad K(a_{r-2}) = \tilde{K}(\tilde{a}_{r-2}), \tag{1.17}$$

$$K(a_{r-1}) = \tilde{K}(\tilde{a}_{r-1})|0, \quad K(a_r) = \tilde{K}(\tilde{a}_r)|1, \tag{1.18}$$

je najkratším binárnym kódovaním abecedy A.

Dôkaz. Dokážeme najprv dve pomocné tvrdenia

Lema 1.1. *Nech $A = \{a_1, a_2, \ldots, a_r\}$ je zdrojová abeceda s pravdepodobnosťami výskytu znakov $p_1 \geq p_2 \geq \cdots \geq p_r$. Potom možno zostrojiť najkratší binárny kód K'' taký, že slová kódové $K''(a_{r-1})$, $K''(a_r)$ sa líšia len v poslednom znaku.*

Dôkaz. Nech K' je najkratšie prefixové binárne kódovanie abecedy A. Ak označíme d'_i dĺžku slova $K'(a_i)$, musí platiť

$$d'_1 \leq d'_2 \leq \cdots \leq d'_r. \tag{1.19}$$

Ak by totiž $d'_i > d'_{i+1}$ vzájomnou zámenou $K'(a_i)$ a $K'(a_{i+1})$ by sme dostali kratšie prefixové kódovanie.

Vytvorme nové kódovanie K'' tak, že $K''(a_i) = K'(a_i)$ pre všetky $i = 1, 2, \ldots, r-1$ a $K''(a_r)$ bude $K'(a_r)$, z ktorého vynecháme posledný znak. K'' vzniklo z najkratšieho prefixového kódovania K', samo však už nemôže byť prefixovým kódovaním, lebo by malo menšiu strednú dĺžku kódového slova než K'. Slovo $K''(a_r)$ musí byť preto prefixom nejakého iného slova $K'(a_j)$. (Je to totiž jediné skrátené

slovo prefixového kódovnia K'). Preto $d'_r - 1 < d'_j$ resp. $d'_r \leq d'_j$, avšak vzhľadom na (1.19) $d'_j \leq d'_r$, z čoho máme $d'_j = d'_r$. Preto

$$d'_1 \leq d'_2 \leq \ldots d'_j = d'_{j+1} = \cdots = d', \tag{1.20}$$

t.j. všetky kódové slová počínajúc $K'(a_j)$ po $K'(a_r)$ majú rovnakú dĺžku, pričom slová $K'(a_j)$, $K'(a_r)$ sa líšia len vo svojom poslednom znaku. Ak sú tieto dve slová posledné, t.j. $j = r - 1$, K' je hľadané kódovanie. Inak z kódovania K' zostrojíme kódovanie K^* tak, že vzájomne zameníme kódy znakov $a_j,\ a_{r-1}$. Exaktne zapísané $K''(a_i) = K'(a_i)$ pre všetky $i = 1, 2, \ldots, r$, $i \neq j, i \neq r - 1$, $K''(a_j) = K'(a_{r-1})$, $K''(a_{r-1}) = K'(a_j)$. □

Lema 1.2. *Nech $\tilde{K}$ je prefixový kód redukovanej abecedy, nech K je kód pôvodnej abecedy $A = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami $p_1, p_2, \ldots, p_r$, pre ktory plati* (1.17), (1.18). *Označme d_i resp. $\tilde{d}_i$ dĺžku slova $K(a_i)$ resp. $\tilde{K}(a_i)$ a $d(K)$ resp. $d(\tilde{K})$ stredné dĺžky kódových slov kódovaní K $\tilde{K}$. Potom platí:*

$$d(K) - d(\tilde{K}) = p_{r-1} + p_r \tag{1.21}$$

Dôkaz. Pretože platí (1.17) je $d_i = \tilde{d}_i$ pre $i = 1, 2, \ldots r-2$. Pretože (1.18), $d_{r-1} = d_r = \tilde{d}_{r-1}+1$. Pre pravdepodobnosti znakov redukovanej abecedy platí $\tilde{p}_i = p_i$ pre $i = 1, 2, \ldots r - 2$, $\tilde{p}_{r-1} = p_{r-1} + p_r$. S využitím týchto vzťahov môžeme písať

$$d(\tilde{K}) = \sum_{i=1}^{r-1} \tilde{p}_i.\tilde{d}_i = p_1.d_1 + p_2.d_2 + \cdots + p_{r-2}.d_{r-2} + (p_{r-1} + p_r).\tilde{d}_{r-1} \tag{1.22}$$

$$d(K) = \sum_{i=1}^{r} p_i.d_i = p_1.d_1 + p_2.d_2 + \cdots + p_{r-2}.d_{r-2} + p_{r-1}.(\tilde{d}_{r-1} + 1) + p_r.(\tilde{d}_{r-1} + 1) \tag{1.23}$$

odkiaľ máme

$$d(K) - d(\tilde{K}) = (p_{r-1} + p_r).(\tilde{d}_{r-1} + 1) - (p_{r-1} + p_r).\tilde{d}_{r-1} = p_{r-1} + p_r. \tag{1.24}$$

□

Konečne dokážeme tvrdenie vety. Nech $\tilde{K}$ je najkratší prefixový kód redukovanej abecedy. Nech kód K je definovaný pomocu vzťahov (1.17) (1.18). Aj kód K je prefixový (lebo $\tilde{K}$ bol prefixový. Podľa lemy 1.1 možno zostrojiť najkratší prefixový kód K'' taký, že kódové slová $K''(a_{r-1},\ K''(a_r)$ sa líšia len v poslednom znaku. Ku kódu K'' možno jednoznačne zostrojiť kód $\tilde{K}''$ kód redukovanej abecedy, takže pr oba kódy platia vzťahy (1.17), (1.18), kde namiesto K píšeme K''. Preto podľa lemy 1.2 platí

$$d(K'') - d(\tilde{K}'') = p_{r-1} + p_r \qquad \text{čiže} \quad d(K'') = d(\tilde{K}'') + p_{r-1} + p_r \tag{1.25}$$

Z tých istých dôvodov môžeme písať

$$d(K) - d(\tilde{K}) = p_{r-1} + p_r \qquad \text{čiže} \quad d(K) = d(\tilde{K}) + p_{r-1} + p_r \tag{1.26}$$

Pretože však $\tilde{K}$ bol najkratší kód redukovanej abecedy, musí byť $d(\tilde{K}) \leq d(\tilde{K}'')$ a preto

$$d(K) = d(\tilde{K}) + p_{r-1} + p_r \leq d(\tilde{K}'') + p_{r-1} + p_r = d(K'') \tag{1.27}$$

Pretože K'' bol najkratší prefixový kód, je

$$d(K'') \leq d(K). \tag{1.28}$$

Z posledných dvoch nerovností máme $d(K) = d(K'')$ – aj kódovanie K je najkratším binárnym kódovaním. □

Algoritmus na zostrojenie Huffmanovho kódu.

Budeme postupne budovať binárny koreňový strom, ktorého listy budú znaky zdrojovej abecedy A. Každý vrchol stromu bude mať priradenú pravdepodobnosť a binárny znak 0 alebo 1

Krok 1: Zostrojme graf $G = (V, H, p)$, kde $V = A$ a kde $p(v)$ je pravdepodobnosť znaku v. Inicializačne položme $H = \emptyset$. Všetky vrcholy z V inicializačne prehlásime za neoznačené.

Krok 2: Nájdeme dva neoznačené vrcholy $u,\ w$ z množiny V s najmenšími pravdepodobnosťami $p(u), p(w)$. Označkujme vrchol u značkou 0, vrchol v značkou 1. Množinu vrcholov V rozšírme o vrchol x, t.j. položme $V := V \cup \{x\}$ pre nejaké $x \notin V$, položme $p(x) := p(u) + p(w)$, $H := H \cup \{(x, u), (x, v)\}$ a nový vrchol x prehlásime za neoznačený.

Krok 3: Ak je graf G súvislý, goto Krok 4, inak goto Krok 2.

Krok 4: Teraz je graf G koreňovým stromom s listami zodpovedajúcimi znakom zdrojovej abecedy A. Všetky vrcholy stromu G okrem koreňa sú označené binárnymi značkami 0 alebo 1. Z koreňa stromu do každého listu vedie jediná cesta, binárne značky vrcholov na tejto ceste určujú prefixový kód každého znaku.

Analogicky sa skonštruuje i n-árny Huffmanov kód. Predpokladajme, že abeceda A má $r = n + k.(n-1)$ znakov. Keby nie doplníme abecedu A o fiktívne znaky s nulovou pravdepodobnosťou – kódové slová priradené týmto fiktívnym znakom zostanú nevyužité. Nájdeme n znakov zdrojovej abecedy s najmenšími pravdepodobnosťami a týmto znakom priradíme znaky kódovej abecedy v ľubovoľnom poradí – budú to posledné znaky ich kódových slov. Abecedu A zredukujeme tak, že namiesto n znakov s najmenšími pravdepodobnosťami dodáme jeden znak so súčtom pravdepodobností nahradených znakov. Zredukovaná abeceda má $n + (k-1).(n-1)$ znakov. Ak $k - 1 > 0$ urobíme znovu redukciu atď.

1.5 Vzťah entropie zdroja a dĺžky najkratšieho kódovania

V tretej kapitole bola definovaná entropia zdroja informácie ako

$$H(\mathcal{Z}) = -\lim_{n\to\infty} \frac{1}{n} \cdot \sum_{(x_1,\ldots,x_n)\in Z} P(x_1, x_2, \ldots, x_n).\log_2(P(x_1, x_2, \ldots, x_n)). \tag{1.29}$$

Pre stacionárny nezávislý zdroj $\mathcal{Z}$ s r-prvkovou abecedou $A = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami znakov $p_1, p_2, \ldots, p_r$ sme ukázali, že $H(\mathcal{Z}) = -\sum_{i=1}^{r} p_1.\log_2(p_i)$.

Majme ľubovoľné binárne prefixové kódovanie K abecedy A s dĺžkami kódových slov $d_1, d_2, \ldots, d_r$ a so strednou dĺžkou slova $d = d(K)$. Chceme zistiť vzťah medzi veličinami $H(\mathcal{Z})$ a d.

$$\begin{aligned}
H(\mathcal{Z}) - d &= \sum_{i=1}^{r} p_i.\log_2\left(\frac{1}{p_i}\right) - \sum_{i=1}^{r} p_i.d_i = \sum_{i=1}^{r} p_i.\left[\log_2\left(\frac{1}{p_i}\right) - d_i\right] = \\
&= \sum_{i=1}^{r} p_i.\left[\log_2\left(\frac{1}{p_i}\right) + \log_2\left(2^{-d_i}\right)\right] = \sum_{i=1}^{r} p_i.\left[\log_2\left(\frac{2^{-d_i}}{p_i}\right)\right] = \frac{1}{\ln 2}.\sum_{i=1}^{r} p_i.\left[\ln\left(\frac{2^{-d_i}}{p_i}\right)\right] \leq \\
&\leq \frac{1}{\ln 2}.\sum_{i=1}^{r} p_i.\left(\frac{2^{-d_i}}{p_i} - 1\right) = \frac{1}{\ln 2}.\left[\sum_{i=1}^{r} 2^{-d_i} - \sum_{i=1}^{r} p_i\right] = \frac{1}{\ln 2}.\left[\sum_{i=1}^{r} 2^{-d_i} - 1\right] \leq 0.
\end{aligned}$$

Prvá nerovnosť v predchádzajúcom postupe vyplýva z nerovnosti $ln(x) \leq x - 1$ aplikovanej na $x = 2^{-d_i}/p_i$, druhá nerovnosť platí preto, lebo prirodzené čísla d_i sú dĺžkami kódových slov prefixového kódovania a platí pre ne Kraftova nerovnosť $\sum_{i=1}^{r} 2^{-d_i} \leq 1$. Platí teda $H(\mathcal{Z}) \leq d(K)$ pre každé prefixové (a teda aj jednoznačne dekódovateľné) kódovanie.

Zvoľme teraz d_i tak, aby platilo $-\log_2(p_i) \leq d_i < -\log_2(p_i) + 1$. Potom

$$\log_2\left(\frac{1}{p_i}\right) \leq d_i \quad \Rightarrow \quad \frac{1}{p_i} \leq 2^{d_i} \quad \Rightarrow \quad 2^{-d_i} \leq p_i$$

Použitím poslednej nerovnosti môžeme písať

$$\sum_{i=1}^{r} 2^{-d_i} \leq \sum_{i=1}^{r} p_i \leq 1$$

Prirodzené čísla d_i pre $i = 1, 2, \ldots, r$ splňujú Kraftovu nerovnosť a preto existuje binárne prefixové kódovanie s dĺžkami kódových slov $d_1, d_2, \ldots, d_r$. Pre strednú dĺžku slova tohoto kódovania platí:

$$d = \sum_{i=1}^{r} p_i.d_i < \sum_{i=1}^{r} p_i.\big[\log_2(p_i) + 1\big] = \sum_{i=1}^{r} p_i.\log_2(p_i) + \sum_{i=1}^{r} p_i = H(\mathcal{Z}) + 1$$

Nech d_{opt} je dĺžka najkratšieho prefixového binárneho kódovania abecedy A. Potom platí $H(\mathcal{Z}) \leq d_{\text{opt}} \leq d \leq H(\mathcal{Z}) + 1$.
Dokázané môžeme zhrnúť do nasledujúcej vety:

Veta 1.4. *Nech $\mathcal{Z}$ je stacionárny nezávislý zdroj s entropiou $H(\mathcal{Z})$, nech d_{opt} je stredná dĺžka kódového slova najkratšieho binárneho prefixového kódovania abecedy A. Potom platí:*

$$H(\mathcal{Z}) \leq d_{\text{opt}} < H(\mathcal{Z}) + 1. \tag{1.30}$$

Príklad 1.4. Majme stacionárny nezávislý zdroj $\mathcal{Z}$ so zdrojovou abecedou $A = \{x, y, z\}$ s troma znakmi, ktorých pravdepodobnosti výskytu sú $p_x = 0.8$, $p_y = 0.1$, $p_z = 0.1$. Kódovanie $K(x) = 0$, $K(y) = 10$, $K(z) = 11$ je najkratšie binárne prefixové kódovanie abecedy A s dĺžkou $d(K) = 1\times0.8+2\times0.1+2\times0.1 = 1.2$. Entropia zdroja $\mathcal{Z}$ je $H(\mathcal{Z}) = 0.922$ bitu na znak. Ak máme dostatočne dlhý N-znakový zdrojový text, potom dĺžku príslušného zakódovaného textu možno odhadnúť číslom $N \times 1.2$, jej dolná hranica hranica určená podľa vety 1.4 je $N \times 0.922$. Dlhý zakódovaný zdrojový text bude teda v tomoto prípade o 30% dlhší ako dolný odhad jeho dĺžky určený entropiou $H(\mathcal{Z})$.

Dal by sa nájsť ešte krikľavejší príklad percentuálnej odchýlky dolnej hranice určenej entropiou zdroja a dĺžkou optimálneho prefixového kódovania (skúste $p_x = 0.98$, $p_y = 0.01$, $p_z = 0.01$). Pretože žiadne jednoznačne dekódovateľné kódovanie zdrojovej abecedy A nemôže mať menšiu strednú dĺžku kódového slova, tento príklad nás nenapĺňa prílišným optimizmom čo sa týka užitočnosti dolného odhadu vo vete 1.4.

Kódovanie znak po znaku však nie je ediným spôsobom, ako zakódovať zdrojový text. V časti **??** bol k zdroju $\mathcal{Z}$ s entropiou $H(\mathcal{Z})$ definovaný zdroj $\mathcal{Z}_{(k)}$ s entropiou $k.H(\mathcal{Z})$, ktorý má za zdrojovú abecedu množinu všetkých k-znakových slov. V prípade, že $\mathcal{Z}$ je stacionárny nezávislý zdroj, je zdroj $\mathcal{Z}_{(k)}$ totožný s produktom $\mathcal{Z}^k$ a je je tiež stacionárnym nezávislým zdrojom. Pre strednú dĺžku $d^{(k)}_{\text{opt}}$ kódového slova najkratšieho binárneho prefixového kódovania abecedy A^k platí vzťah (1.30) z vety 1.4:

$$\begin{aligned} H(\mathcal{Z}_{(k)}) \leq & d^{(k)}_{\text{opt}} < H(\mathcal{Z}_{(k)}) + 1 \\ k.H(\mathcal{Z}) \leq & d^{(k)}_{\text{opt}} < k.H(\mathcal{Z}) + 1 \\ H(\mathcal{Z}) \leq & \frac{d^{(k)}_{\text{opt}}}{k} < H(\mathcal{Z}) + \frac{1}{k} \end{aligned} \tag{1.31}$$

Práve dokázané poznatky zhrnieme v nasledujúcej vete:

Veta 1.5. Základná veta o kódovaní zdrojov. *Nech $\mathcal{Z} = (A^*, P)$ je stacionárny nezávislý zdroj s entropiou $H(\mathcal{Z})$. Potom je stredná dĺžka zakódovaného binárneho textu pripadajúca na jeden znak zdrojovej abecedy A zdola ohraničená entropiou $H(\mathcal{Z})$. Pritom sa dá nájsť prirodzené číslo k a binárne prefixové kódovanie slov z A^k také, že stredná dĺžka zakódovaného textu pripadajúca na jeden znak zdrojovej abecedy A je ľubovoľne blízko entropii $H(\mathcal{Z})$.*

Základná veta o kódovaní zdrojov platí aj pre ľubovoľný stacionárny zdroj $\mathcal{Z}$ (dôkaz je však o čosi zložitejší). Jej význam je v tom, že ukazuje entropiu zdroja ako limitnú hodnotu strednej dĺžky binárne zakódovaného textu pripadajúceho na jeden znak zdrojovej abecedy. Ukazuje sa tu, že pojem entropie bol dobre zvolený a má svoj hlboký význam. Všimnime si tiež, že pre binárne kódovanie vo vzťahu (1.30) vo vete 1.4 vystupuje entropia $H(\mathcal{Z})$ bez prepočítavacieho koeficientu (resp. s koeficientom 1), čo je dôsledok toho, že sme šťastne zvolili číslo 2 za základ logaritmu pri Shannonovej definícii informácie, resp. pri Shannonovej – Hartleyovej formuli pre entropiu.

Ako sme už ukázali, prirodzený jazyk ani zďaleka nemožno považovať za nezávislý zdroj, jeho entropia je oveľa menšia, ako entropia prvého písmena $H_1 = -\sum_i p_i \log_2(p_i)$. V takýchto prípadoch by sme mohli dostať kratšiu dĺžku zakódovanej správy tak, že by sme za znaky zdrojovej abecedy brali dvojice, trojice, prípadne n-tice pôvodnej abecedy A. Huffmanove kódovanie je základom mnohých metód kompresie údajov, kde sa k správe zakódovanej v blokovom binárnom kóde hľadá efektívnejší spôsob uloženia.

1.6 Kódy objavujúce chyby

V tejto časti sa budeme zaoberať blokovými kódmi s n-prvkovou kódovou abecedou a mnohokrát špeciálne dekadickými číslami. Do spracovanie takýchto prirodzených kódov je býva včlenený aj ľudský činiteľ, ktorý je zdrojom častých chýb. Otázkou teraz je, či je možné nájsť kód taký, ktorý by zistil, že pri prenose nastala jedna, alebo aj viac chýb istého druhu. Z anglosaskej literatúry máme údaje o relatívnej početnosti chýb vznikajúcich písaním textov na klávesnici resp. písacom stroji.

- Jednoduchá chyba $a \to b$ 79%
- Susedná transpozícia $ab \to ba$ 10.2%
- Skoková transpozícia $abc \to cba$ 0.8%
- Blíženci $aa \to bb$ 0.6%
- Fonetická chyba $X0 \to 1X$ 0.5%
- Ostatné chyby 8.9%

Vidíme, že najbežnejšie ľudské chyby sú jednoduchá chyba a zámena poradia dvoch susedných písmen. Medzi ostatnými chybami môže byť aj vynechanie či pridanie znaku, avšak blokový kód takúto chybu ihneď odhalí, lebo mení dĺžku kódového slova. Fonetická chyba je pravdepodobne anglickou špecialitou a vychádza z malého rozdielu medzi anglickými číslovkami (napr. fourteen – fourty, fifteen – fifty apod.).

Ak má kódová abeceda B n znakov, potom počet všetkých slov dĺžky k je n^k – to je najväčší možný počet slov blokového kódu dĺžky k s n kódovými znakmi. Jediným spôsobom, ako na strane prijímača zistiť chybu v prijatom slove je tento: Pre **kódové slová** využiť len časť z n^k možných slov, ostatné slová prehlásiť za **nekódové**. Ak prijmeme nekódové slovo, vieme, že sme prijali slovo s chybou. Problémom však je, ako určiť množinu kódových slov tak, aby pri jednej (alebo i vac) dovolenej chybe istého druhu vzniklo z kódového slova nekódové a ako rýchlo určiť, či prijaté slov je alebo nie je kódové. Pri študovaní tejto problematiky sa najskôr obmedzíme na jednoduché chyby, ktorým niekedy hovorím preklepy.

Definícia 1.5. Hammingova vzdialenosť $d(\mathbf{v}, \mathbf{w})$ dvoch slov $\mathbf{v} = v_1 v_2 \dots v_n$, $\mathbf{w} = w_1 w_1 \dots w_n$ je počet miest, na ktorých sa znaky slov $\mathbf{v}$, $\mathbf{w}$ líšia, t.j.

$$d(\mathbf{v}, \mathbf{w}) = \big|\{i \mid v_i \neq w_i, \quad i = 1, 2, \dots, n\}\big|. \tag{1.32}$$

Minimálna vzdialenosť ΔK blokového kódu K je minimum zo vzdialeností všetkých dvojíc slov kódu K. Hovoríme, že kód K **objavuje t-násobné jednoduché chyby**, ak pri zmene ľubovoľných t znakov kódového slova $\mathbf{u}$ vznikne nekódové slovo. Ak teda prijmeme nekódové slovo, hovoríme, že sm objavili chybu. Všimnime si, že blokový kód K s minimálnou vzdialenosťou $\Delta K = d$ objavuje $(d-1)$-násobné jednoduché chyby.

Príklad 1.5. [Kód dva z piatich] Dva prvky z piatich možno vybrať $\binom{5}{2} = 10$ spôsobmi, čo možno využiť pre kódovanie dekadických cifier nasledovne:

1	11000	6	00101
2	10100	7	00011
3	10010	8	00110
4	10001	9	01100
5	01001	0	01010

Kód dva z piatich objavuje jednu chybu – pri zmene ktorejkoľvek 0 na 1 vznikne nekódové slovo s troma znakmi 1, pri zmene 1 na 0 dostaneme nekódové slovo obsahujúce len jeden znak 1. Kódové slová 11000 a 10100 majú však Hammingovu vzdialenosť rovnú 2, z čoho vyplýva, že kód dva z piatich neobjavuje všetky 2-násobné jednoduché chyby.

Príklad 1.6. Kód s kontrolou parity je osembitový kód, kde prvých 7 bitov je ľubovoľný 7-miestny binárny blokový kód a kde je posledný binárny znak doplnený tak, aby počet jednotkových bitov bol párny. Kód s kontrolou parity objavuje jednu jednoduchú chybu, jeho minimálna vzdialenosť je 2. Princíp kontroly paritou bol veľmi často používaný pri prenosoch a bočas sa s ním stretneme aj v súčasnosti.

Príklad 1.7. Zdvojovací kód. Ide o kód párnej dĺžky, v ktorom sa každý znak opakuje dvakrát. Zdvojovací binárny kód dĺžky 6 má osem kódových slov:

000000 000011 001100 001111 110000 110011 111100 111111

Zdvojovací kód má minimálnu vzdialenosť rovnú 2, objavuje jednu jednoduchú chybu.

Príklad 1.8. Opakovací kód. Princíom opakovacieho kódu je niekoľkonásobné opakovanie toho istého znaku. Kódové slová sú len slová pozostávajúce z toho istého znaku – napr 11111, 22222, ... , 99999, 00000. Opakovací kód K dĺžky n má minimálnu vzdialenosť $\Delta K = n$ a preto objavuje $(n-1)$-násobné jednoduché chyby. Všimnime si, že za predpokladu, že nastali maximálne dve chyby, pri opakovacom kóde dĺžky 5 vieme zrekonštruovat pôvodné slovo. Ak prijmeme 10191, za predpokladu maximálne dvoch chýb vieme, že bolo vyslané slovo 11111.

Príklad 1.9. Medzinárodné číslo vagónu je 12-miestne dekadické číslo tvaru

$$X\ X\ XX\ X\ XXX\ XXX\ X$$

Prvá cifra predstavuje číslo medzinárodného spoločenstva, druhá triedu vyhovovania medzinárodným predpisom, v tretej a štvrtej cifre je zakdóvaný vlastník, piata cifra obsahuje kód základného triedenia vozňa, ďalšie trojčíslie predstavuje kód technickej špecifikácie vozňa, trojčísle od deviatej po jedenástu cifru obsahuje poradové číslo vozňa a posledná dvanásta cifra je kontrolná číslica.
Majme číslo vagóna

$$a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12}$$

Kontrolná číslica a_{12} sa určí tak, aby ciferný súčet čísel

$$2a_1\ a_2\ 2a_3\ a_4\ 2a_5\ a_6\ 2a_7\ a_8\ 2a_9\ a_{10}\ 2a_{11}\ a_{12}$$

bol deliteľný číslom 10. Cifry na párnych a nepárnych miestach sa spracovávajú odlišne – evidentne tu vidieť snahu poistiť sa i proti susednej zámene. Nech na dvoch susedných miestach sú cifry C, D, nech C je na nepárnom mieste. Označme $\delta(Y)$ ciferný súčet čísla $2Y$ pre $Y = 0, 1, \ldots, 9$. Potom

$$\delta(Y) = \begin{cases} 2Y & \text{ak } Y \leq 4 \\ 2Y - 9 & \text{ak } Y > 4 \end{cases}$$

Hľadajme, pre ktoré hodnoty cifier C, D sa kontrolná číslica nezmení. Aby sa kontrolná číslica pri susednej zámene cifier nezmenila, musí dať súčet $\delta(C) + D$ ten istý zvyšok pri delení desiatimi ako $\delta(D) + C$ a teda ich rozdiel musí byť deliteľný desiatimi.

$$\delta(C) + D - \delta(D) - C = \begin{cases} 2C + D - 2D - C = C - D & \text{ak } C \leq 4 \text{ a } D \leq 4 \\ 2C - 9 + D - 2D - C = C - D - 9 & \text{ak } C \geq 5 \text{ a } D \leq 4 \\ 2C + D - 2D + 9 - C = C - D + 9 & \text{ak } C \leq 4 \text{ a } D \geq 5 \\ 2C + 9 + D - 2D - 9 - C = C - D & \text{ak } C \geq 5 \text{ a } D \geq 5 \end{cases}$$

V prvom a štvrtom prípade rozdiel $C - D$ je deliteľný desiatimi práve vtedy, keď $C = D$, z čoho vyplýva, že kód rozpozná každú susednú zámenu takých cifier, že sú obe menšie než 5 alebo obe väčšie než 4.
V druhom prípade ak $C \geq 5$ a súčasne $D \leq 4$ potom $1 \leq (C - D) \leq 9$. Výraz $\delta(C) + D - \delta(D) - C$ je v tomto prípade rovný $(C - D) - 9$. Posledný výray môže byť deliteľný desiatimi len vtedy, keď $C - D = 9$ čo môže nastať len pre dvojicu $C = 9$, $D = 0$.
V treťom prípade ak $C \leq 4$ a $D \geq 5$ je $0-9 = -9 \leq (C-D) \leq 4-5 = -1$. Výraz $\delta(C)+D-\delta(D)-C$ je v tomto prípade rovný $(C - D) + 9$. Posledný výraz môže byť deliteľný desiatimi len vtedy, keď $C - D = -9$ čo môže nastať len pre dvojicu $C = 0$, $D = 9$. Vidíme, že rovnica

$$\delta(C) + D - \delta(D) - C \equiv 0 \mod 10$$

má len dve riešenie $C = 0$, $D = 9$ a $C = 9$, $D = 0$. Kódovanie vozňov teda objaví jednu jednoduchú chybu alebo jednu susednú zámenu dvojice znakov rôznej od 09 a 90. Ak sa v čísle vozňa kdekoľvek zamení dvojica 09 za 90, resp. 90 za 09, kontrolná číslica sa nezmení a teda kód takúto chybu nezistí. Objavovanie ktorejkoľvek susednej zámeny sa však konštruktérom tohoto kódu nepodarilo zaistiť.

Príklad 1.10. ISBN – International Standrard Book Number je 10 miestne číslo prideľovné každej oficiálne vydanej knihe, v ktorom prvé štyri znaky $a_1a_2a_3a_4$ určujú krajinu a vydavatľstvo, ďalších päť znakov $a_5a_6a_7a_8a_9$ predstavuje číslo knihy v rámci špecifikovaného vydavateľstva a posledný znak a_{10} je kontrolný znak určený rovnicou

$$a_{10} \equiv \sum_{i=1}^{9} i.a_i \mod 11 \tag{1.33}$$

Posledná rovnica je totožná s

$$\sum_{i=1}^{10} i.a_i \equiv \mod 11, \tag{1.34}$$

pretože $-a_{10} \equiv -a_{10} + 11.a_{10} \equiv 10.a_{10} \mod 10$. Ak $a_{10} = 10$, píše sa na mieste a_{10} znak X. Toto je istá nevýhoda ISBN kódovania, pretože kódová abeceda je 11-prvková, ale znak X sa využíva len zriedka. ISBN kód objavuje všetky jednoduché chyby. Nech sa znak a_i zamení znakom a_i' a nech

$i.a_i \equiv i.a_i'$ mod 11. Potom $i.(a_i - a_i') \equiv 0$ mod 11, t.j. $i.(a_i - a_i')$ je deliteľné číslom 11, čo je pre $i = 1, 2, \ldots, 10$ možné len ak $(a_i - a_i') = 0$.

Skúmajme, či ISBN kód odhalí všetky susedné zámeny. Nech $x,\ y$ je dvojica znakov ktorých susednú zámenu na miestach $i,\ i+1$ ISBN kód neodhalí potom $x,\ y$ sú riešením rovnice

$$i.x + (i+1)y \equiv i.y + (i+1)x \mod 11$$

Túto rovnicu môžeme ďalej upravovať

$$\begin{aligned} i.x + (i+1)y - i.y - (i+1)x &\equiv 0 \mod 11 \\ y - x &\equiv 0 \mod 11, \end{aligned}$$

čo je pre $x,\ y$ z intervalu $\langle 0, 10\rangle$ možné len ak $x = y$. ISBN kód odhaľuje susedné zámeny.

Príklad 1.11. EAN European Article Number je 13-miestny dekadický kód, ktorým sa jedinečne označujú výrobky v rámci Európy. EAN kód býva prevedený do čiarového kódu, ktorý je umiestnený na obale výrobku. Pri manipulácii s výrobkami sa tento kód sníma opticky, čím sa znižuje pracnosť pri evidencii, fakturácii, inventarizácii a ďalších operáciách s výrobkom.

Prvých dvanásť znakov a_1 až a_{12} kódu EAN je významových, znak a_{13} je kontrolný a vypočíta sa z rovnice

$$a_{13} \equiv -(1.a_1 + 3.a_2 + 1.a_3 + 3.a_4 + \cdots + 1.a_{11} + 3.a_{12}) \mod 11$$

Kód EAN odhaľuje jednoduché chyby. Pre dvojicu znakov $x,\ y$, na dvoch susedných miestach nepárnom a párnom kód neodhalí susednú zámenu, ak

$$\begin{aligned} (x + 3y) - (3x + y) &\equiv 0 \mod 10 \\ (2y - 2x) &\equiv 0 \mod 10 \\ 2.(y - x) &\equiv 0 \mod 10 \end{aligned}$$

Posledná rovnica má tieto riešenia $x,\ y$:

$$\begin{aligned} &(0,0),\ (0,5),\ (1,1),\ (1,6),\ (2,2),\ (2,7),\ (3,3),\ (3,8),\ (4,4),\ (4,9), \\ &(5,5),\ (5,0),\ (6,6),\ (6,1),\ (7,7),\ (7,2),\ (8,8),\ (8,3),\ (9,9),\ (9,4) \end{aligned}$$

Pre desať dvojíc susedných znakov kód EAN neobjaví susednú zámenu. Z hľadiska počtu neobjavených chýb je na tom horšie ako medzinárodný číselník vozňov, pre ktorý je neobjaviteľní iba susedná zámena znakov 0 a 9.

1.7 Elementárne metódy objavovania chýb

1.7.1 Kódy s kontrolnou rovnicou mod 10

Pre dekadické kódy určíme kontrolnú číslicu a_n z rovnice

$$a_n \equiv -\sum_{i=1}^{n-1} w_i.a_i \mod 10. \tag{1.35}$$

Tento prístup možno ešte trochu zovšeobecniť tak, že kódové slová tvorené znakmi a_1 až a_n sú práve tie slová, ktoré vyhovujú tzv. kontrolnej rovnici

$$\sum_{i=1}^{n} w_i.a_i \equiv c \mod 10. \tag{1.36}$$

Ak sa v slove $a_1a_2\ldots a_n$ zmení a_j na a'_j, bude ľavá strana kontrolnej rovnice pre takto zmenené slovo rovná

$$\sum_{i=1}^{n} w_i.a_i + w_j.a'_j - w_j.a_j \equiv c + w_j.(a'_j - a_j) \mod 10$$

Príslušný kód neobjaví jednoduchú chybu, ak

$$w_j.(a'_j - a_j) \equiv 0 \mod 10$$

Posledná rovnica len riešenia $a'_j = a_j$ práve vtedy, keď w_j nie je súdeliteľné s číslo 10. Na miestach w_i môžu byť len čísla 1, 3, 7 a 9.

Skúsme zistiť, či kód s kontrolnou rovnicou (1.36) objavuje susedné zámeny. Kód nezistí susednú zámenu znakov $x,\ y$ na miestach $i,\ i+1$ práve vtedy, keď

$$\begin{aligned} w_i.y + w_{i+1}.x - w_i.x - w_{i+1}.y &\equiv 0 \mod 10 \\ w_i.(y-x) - w_{i+1}.(y-x) &\equiv 0 \mod 10 \\ (w_i - w_{i+1})(y-x) &\equiv 0 \mod 10 \end{aligned}$$

K tomu, aby posledná rovnica nemala okrem riešení $x = y$ žiadne ďalšie je nutné a stačí aby $(w_i - w_{i+1})$ bolo nesúdeliteľné s 10. Ak má však kód s kontrolnou rovnicou (1.36) rozoznávať jednoduché chyby, musí byť $w_i \in \{1,3,7,9\}$ a preto je $(w_i - w_{i+1})$ vždy párne.

Veta 1.6. *Nech K je desiatkový blokový kód dĺžky n s kontrolnou rovnicou* (1.36)*. Kód K objavuje jednoduché chyby práve vtedy, keď sú všetky w_i nesúdeliteľné s 10, t.j. $w_i \in \{1,3,7,9\}$. Žiaden desiatkový blokový kód dĺžky n s kontrolnou rovnicou* (1.36) *neobjavuje jednoduché chyby a súčasne aj susedné zámeny.*

1.7.2 Kontrola modulo 11

Tieto kódy pracujú s kódovou abecedou $B \cup \{X\}$, kde $B = \{1,2,3,4,5,6,7,8,9,0\}$, pričom kódové slová majú všetkých prvých $n-1$ znakov z abecedy B posledný kontrolný znak a_n z abecedy $B \cup \{X\}$ určený tak, aby platila nasledujúca kontrolná rovnica

$$\sum_{i=1}^{n} w_i.a_i \equiv c \mod 11. \tag{1.37}$$

Podobne ako v prípade kontroly modulo 10 ukážeme, že kód objavuje jednoduché chyby na mieste j práve vtedy, keď rovnica

$$w_j.(a'_j - a_j) \equiv 0 \mod 11 \tag{1.38}$$

nemá okrem $a'_j = a_j$ žiadne iné riešenia a to je práve vtedy, keď w_j je nesúdeliteľné s 11 na čo stačí, aby $w_j \neq 0$.
Na to, aby kód s kontrolou modulo 11 objavoval susedné zámeny na miestach $i,\ i+1$ stačí, aby rovnica

$$(w_i - w_{i+1}).(y-x) \equiv 0 \mod 11 \tag{1.39}$$

okrem riešení, kde $x = y$ nemala žiadne iné riešenia. Na to však stačí, aby $w_i \neq w_{i+1}$. Príkladom tohoto kódu je kód ISBN.
Nakoniec poznamenajme, že vlastnosť objavovania jednoduchých chýb a susedných zámen sa nestratí ak dovolíme, aby všetky znaky kódových slov boli z abecedy $B \cup \{X\}$.

Mnoho dobrých vlastností má tzv. **geometrický kód modulo 11**, kde čísla w_i v kontrolnej rovnici (1.37) sú určené ako

$$w_i = 2^i \mod 11 \tag{1.40}$$

1.8 Kódovanie s kontrolným znakom nad grupou

Pri kódoch s kontrolnou rovnicou modulo 10 alebo 11 kódy objavovali jednoduché chyby práve vtedy, keď zobrazenie $\delta(a_i) = R(w_i.a_i \mod 10)$ (zvyšok po delení čísla $w_i.a_i$ desiatimi) bolo permutáciou. Dokonca aj priradenie $\delta(x) =$ ciferný súčet 2.x v kódovaní železničných vozňov je permutáciou a tam sme dosiahli doteraz najlepší výsledok, čo sa týka zabezpečenia proti susedným zámenám. Vzniká teda myšlienka členy $w_i.a_i$ kontrolnej rovnice nahradiť permutáciami $\delta_i(a_i)$.

Príklad 1.12. Medzinárodné číslo vozňa je vlastne kód s permutáciami

$$\delta_1 = \delta_3 = \cdots = \delta_{11} := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 0\ 2\ 4\ 6\ 8\ 1\ 3\ 5\ 7\ 9 \end{pmatrix}$$
$$\delta_2 = \delta_4 = \cdots = \delta_{12} := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \end{pmatrix}$$

a s kontrolnou rovnicou

$$\sum_{i=1}^{12} \delta_i(a_i) \equiv 0 \mod 10$$

Príklad 1.13. Kód nemeckých poštových poukážok je desaťmiestny dekadický kód $a_1a_2 \ldots a_{10}$ s kotrolným znakom a_{10} s kontrolnou rovnicou

$$\sum_{i=1}^{10} \delta_i(a_i) \equiv 0 \mod 10$$

kde

$$\delta_1 = \delta_4 = \delta_7 := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 0 \end{pmatrix} \qquad \delta_2 = \delta_5 = \delta_8 := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 2\ 4\ 6\ 8\ 0\ 1\ 3\ 5\ 7\ 9 \end{pmatrix}$$
$$\delta_3 = \delta_6 = \delta_9 := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 3\ 6\ 9\ 1\ 4\ 7\ 0\ 2\ 5\ 8 \end{pmatrix} \qquad \delta_{10} := \begin{pmatrix} 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9 \\ 0\ 9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{pmatrix}$$

Ani jeden z uvedených kódov neobjavuje všetky susedné zámeny. Preto ďalším zovšeobecnením je nahradenie grupy zvyškových tried s grupovou operáciou $+ \mod 10$ nejakou inou grupou $\mathbb{G} = (A, *)$ a kontrolnú rovnicu formulovať ako

$$\prod_{i=1}^{n} \delta_i(a_i) = c \tag{1.41}$$

Multiplikatívny tvar grupovej operácie $*$ naznačuje, že grupa $\mathbb{G}$ nemusí byť komutatívna.

Definícia 1.6. Nech A je abeceda, nech $\mathbb{G} = (A, *)$ je grupa. Nech $\delta_1, \delta_2, \ldots, \delta_n$, sú permutácie na A. Potom kontrolnou rovnicou (1.41) definovaný kód nazveme **kód s kontrolným znakom nad grupou** $\mathbb{G}$.

Permutácie sú vzájomne jednoznačné zobrazenia A na A. Preto ku každej permutácii δ existuje inverzná permutácia označovaná δ^{-1} pre ktorú platí

$$\delta(a) = x \quad \text{práve vtedy, keď} \quad \delta^{-1}(x) = a.$$

Ak máme dve permutácie δ_i, δ_j potom predpisom $\delta_i\big(\delta_j(a)\big)$ vznikne permutácia, ktorú budeme značiť $\delta_i \circ \delta_j$, teda

$$\delta_i \circ \delta_j(a) = \delta_i\big(\delta_j(a)\big) \quad \forall a \in A \tag{1.42}$$

Veta 1.7. *Aby kód K s kontrolným znakom nad grupou $\mathbb{G} = (A, *)$ rozpoznal zámenu ľubovoľných susedných znakov na miestach i, $i+1$ je nevyhnutné a stačí, aby*

$$x * \delta_{i+1} \circ \delta_i^{-1}(y) \neq y * \delta_{i+1} \circ \delta_i^{-1}(x) \tag{1.43}$$

pre všetky $x \in A$, $y \in A$, $x \neq y$.

Pre Abelovskú grupu $\mathbb{G} = (A, +)$ možno vzťah (1.43) prepísať v tvare $x + \delta_{i+1} \circ \delta_i^{-1}(y) \neq y + \delta_{i+1} \circ \delta_i^{-1}(x)$, odkiaľ máme nasledujúci dôsledok:

Dôsledok. *Kód K s kontrolným znakom nad Abelovou grupou $\mathbb{G} = (A, +)$ objavuje zámenu ľubovoľných susedných znakov na miestach i, $i+1$ práve vtedy, keď pre ľubovoľné $x,\ y \in A$, $x \neq y$ platí:*

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) \neq y - \delta_{i+1} \circ \delta_i^{-1}(y). \tag{1.44}$$

Dôkaz. Nech kód K rozpoznáva susednú zámenu na miestach i, $i+1$. Potom pre ľubovoľné a_i, a_{i+1} také, že $a_i \neq a_{i+1}$ platí:

$$\delta_i(a_i) * \delta_{i+1}(a_{i+1}) \neq \delta_i(a_{i+1}) * \delta_{i+1}(a_i) \tag{1.45}$$

Pre ľubovoľné $x \in A$ existuje nejaké $a_i \in A$ také, že , potom $a_i = \delta_i^{-1}(x)$. Podobne pre ľubovoľné $y \in A$ existuje nejaké $a_{i+1} \in A$ také, že , potom $a_{i+1} = \delta_i^{-1}(y)$. Dosaďme do (1.45) najprv x za $\delta_i(a_i)$ a y za $\delta_i(a_{i+1})$, potom $\delta_i^{-1}(x)$ za a_i a $\delta_i^{-1}(y)$ za a_{i+1}. Dostaneme

$$\begin{aligned} x * \delta_{i+1}(a_{i+1}) &\neq y * \delta_{i+1}(a_i) \\ x * \delta_{i+1}\big(\delta_i^{-1}(y)\big) &\neq y * \delta_{i+1}\big(\delta_i^{-1}(x)\big) \\ x * \delta_{i+1} \circ \delta_i^{-1}(y) &\neq y * \delta_{i+1} \circ \delta_i^{-1}(x) \end{aligned}$$

Nech platí (1.45) pre všetky $x,\ y \in A$, $x \neq y$. Potom (1.45) platí aj pre $x = \delta_i(a_i)$, $y = \delta_i(a_{i+1})$, kde $a_i,\ a_{i+1} \in A$, $a_i \neq a_{i+1}$.

$$\begin{aligned} \delta_i(a_i) * \delta_{i+1} \circ \delta_i^{-1}(\delta_i(a_{i+1})) &\neq \delta_i(a_{i+1}) * \delta_{i+1} \circ \delta_i^{-1}(\delta_i(a_i)) \\ \delta_i(a_i) * \delta_{i+1}\Big(\underbrace{\delta_i^{-1}\big(\delta_i(a_{i+1})\big)}_{a_{i+1}}\Big) &\neq \delta_i(a_{i+1}) * \delta_{i+1}\Big(\underbrace{\delta_i^{-1}\big(\delta_i(a_i)\big)}_{a_i}\Big) \\ \delta_i(a_i) * \delta_{i+1}(a_{i+1}) &\neq \delta_i(a_{i+1}) * \delta_{i+1}(a_i) \end{aligned}$$

z čoho vyplýva, že kód K objavuje susednú zámenu na miestach i, $i+1$. □

Všimnime si vzťah (1.44). Ten hovorí, že priradenie $x \mapsto \big(x - \delta_{i+1} \circ \delta_i^{-1}(x)\big)$ je prosté – je tiež permutáciou.

Definícia 1.7. Permutácia δ (multiplikatívnej) grupy $\mathbb{G} = (A, *)$ sa nazýva **úplným zobrazením**, ak

$$\eta(x) = x * \delta(x) \tag{1.46}$$

je zase permutácia. Permutácia δ (aditívnej) grupy $\mathbb{G} = (A, +)$ sa nazýva **úplným zobrazením**, ak

$$\eta(x) = x + \delta(x) \tag{1.47}$$

je zase permutácia.

Veta 1.8. *Kód K s kontrolným znakom nad Ábelovou grupou $\mathbb{G} = (A, +)$ objavujúci jednoduché chyby a susedné zámeny existuje práve vtedy, keď existuje úplné zobrazenie grupy $\mathbb{G}$.*

Dôkaz. Definujme zobrazenie $\mu : A \to A$ predpisom $\mu(x) = -x$. Zobrazenie μ je prosté - je to permutácia. Pre ľubovoľnú permutáciu δ množiny A je zobrazenie $x \mapsto -\delta(x) = \mu \circ \delta(x)$ zase permutáciou.

Nech K objavuje susedné zámeny, potom podľa dôsledku vety 1.7 je zobrazenie $x \mapsto \big(x - \delta_{i+1} \circ \delta_i^{-1}(x)\big)$ permutáciou. Ale

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) = x + \underbrace{\mu \circ \delta_{i+1} \circ \delta_i^{-1}(x)}_{\delta(x)} = x + \delta(x)$$

Permutácia δ definovaná predpisom $\delta = \mu \circ \delta_{i+1} \circ \delta_i^{-1}$ je hľadaným úplným zobrazením.

Nech existuje úplné zobrazenie δ grupy $\mathbb{G}$. Definujme

$$\delta_i = (\mu \circ \delta)^i. \tag{1.48}$$

Potom

$$x - \delta_{i+1} \circ \delta_i^{-1}(x) = x - (\mu \circ \delta)^{i+1} \circ (\mu \circ \delta)^{-i}(x) = x - (\mu \circ \delta)(x) = x + \delta(x),$$

z čoho vyplýva, že $x - \delta_{i+1} \circ \delta_i^{-1}(x)$ je permutácia. Podľa dôsledku vety 1.7 kód s kontrolným znakom nad grupou $\mathbb{G}$ s permutáciami δ_i definovanými v (1.48) objavuje susedné zámeny. □

Veta 1.9. *Nech $\mathbb{G}$ je Abelova konečná grupa. Potom platí:*

a) Ak je $\mathbb{G}$ grupa nepárneho rádu, potom je identita na $\mathbb{G}$ úplným zobrazením.

b) Grupa $\mathbb{G}$ rádu $r = 2.m$, kde m je nepárne číslo, nemá žiadne úplné zobrazenie

c) Nech $\mathbb{G} = (A, +)$ je Abelova grupa párneho rádu. Potom na $\mathbb{G}$ existuje úplné zobrazenie práve vtedy, keď grupa obsahuje aspoň dve rôzne involúcie, t.j. také prvky $g \in A$, že $g \neq 0$, a $g+g = 0$

Dôkaz. Dôkaz tejto vety patrí do teórie konečných grúp, presahuje zámery tejto publikácie, preto vetu uvádzame bez dôkazu.

Ak máme abecedu $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ na ktorej definujeme grupovú operáciu $\oplus$ predpisom $a \oplus b = (a + b) \mod 10$, možno všetky kódy s kontrolnou cifrou modulo 10 interpretovať ako kódy s kontrolným znakom nad Abelovou grupou $\mathbb{G} = (A, \oplus)$. Táto grupa je rádu $r = 2.5$ a preto v nej neexistuje úplné zobrazenie.

Dôsledok. Neexistuje žiaden dekadický kód s kontrolným znakom nad Abelovou grupou $\mathbb{G} = (A, \oplus)$, kde $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, ktorý by objavoval jednoduché chyby a susedné zámeny.

Jediná šanca na zostrojenie dekadického kódu, ktorý by objavoval jednoduché chyby a susedné zámeny je skúsiť kód s kontrolným znakom nad nekomutatívnou grupou.

Definícia 1.8. Diederovská grupa $\mathbb{D}_n$ j konečná grupa rádu $2.n$ tvaru

$$\big\{1, a, a^2, \ldots, a^{n-1}, b, ab, a^2.b, \ldots, a^{n-1}b\big\}, \tag{1.49}$$

Kde platí

$$a^n = 1 \quad (a^i \neq 1 \text{ pre } i = 1, 2, \ldots, n-1 \tag{1.50}$$

$$b^2 = 1 \quad (b \neq 1) \tag{1.51}$$

$$b.a = a^{n-1}.b \tag{1.52}$$

Diederovu grupu $\mathbb{D}_n$ budem značiť

$$\mathbb{D}_n = \big\langle a, b \;\big|\; a^n = 1 = b^2,\ ba = a^{n-1}b \big\rangle \tag{1.53}$$

Diedrovskú grupu $\mathbb{D}_n$ možno interpretovať ako grupu symetrií pravidelného n-uholníka – a ako rotáciu okolo stredu o uhol $2\pi/n$, b ako osovú súmernosť. Pre $\mathbb{D}_3$ $1 = (ABC)$, $a = (CAB)$, $a^2 = (BCA)$, $b = (ACB)$, $ab = (BAC)$, $a^2b = (CBA)$.

Veta 1.10. *Nech* $\mathbb{D}_n = \langle a, b \mid a^n = 1 = b^2,\ ba = a^{n-1}b \rangle$ *je Diedorva grupa nepárneho rádu* n, $n \geq 3$. *Definujme permutáciu* $\delta : \mathbb{D}_n \to \mathbb{D}_n$ *predpisom*

$$\delta(a^i) = a^{n-1-i} \quad a \quad \delta(a^i b) = a^i b \quad \forall i = 1, 2, \ldots, n-1 \tag{1.54}$$

Potom pre permutáciu δ *platí:*

$$x.\delta(y) \neq y.\delta(x) \quad \forall x, y \in \mathbb{D}_n \text{ také, že } x \neq y. \tag{1.55}$$

Dôkaz. Prv než sa pustíme do samotného dôkazu uvedomme si jednu skutočnosť. Podľa definície Diederovej grupy je $b.a = a^{n-1}b$. Keďže $a^{n-1}.a = 1$, je $a^{n-1} = a^{-1}$, a preto platí $ba = a^{-1}b$. Nech k je ľubovoľné prirodzené číslo. Potom $b.a^k = a^{-1}ba^{k-1} = a^{-2}ba^{k-2} = \cdots = a^{-k}b$. Pre ľubovoľné celé číslo platí

$$b.a^k = a^{-k}b. \tag{1.56}$$

Ľahko sa overí, že δ je prosté zobrazenie – permutácia. Aby sme overili (1.55), budeme rozoznávať tri prípady.
1. prípad: $x = a^i$, $y = a^j$, kde $i \neq j$, $0 \leq i, j \leq n-1$. Keby platilo $x.\delta(y) = y.\delta(x)$, potom by $a^i.a^{n-1-j} = a^j.a^{n-1-i}$, z čoho vyplýva $a^{2i-2j} = a^{2(i-j)} = 1$. Číslo $2(i-j)$ musí byť delideľné napárnym číslom n, keby totiž $2(i-j) = kn+r$, kde $1 \leq r \leq n-1$, potom by $a^{2(i-j)} = a^{kn+r} = a^{kn}a^r = 1.a^r \neq 1$. Ak má nepárne n deliť $2(i-j)$, musí byť $(i-j)$ deliteľné číslom n, čo môže nastať len tak, že $(i-j) = 0$, lebo $0 \leq i, j \leq n-1$.
2. prípad: $x = a^i$, $y = a^j b$, $0 \leq i, j \leq n-1$. Nech $x.\delta(y) = y.\delta(x)$, t.j. $a^i a^j b = a^j b a^{n-1-i}$. Použitím (1.56) máme $a^{i+j}b = a^j.a^{i+1}b$ odkiaľ postupne dostaneme $a^{i+j} = a^{i+j+1}$, $1 = a$. V definícii Diederovej grupy $\mathbb{D}_n$ pre $n \geq 3$ však $a \neq 1$.

3. prípad: $x = a^i b$, $y = a^j b$, $0 \leq i, j \leq n-1$. Nech $x.\delta(y) = y.\delta(x)$ čo v tomto prípade znamená $a^i b.a^j b = a^j b a^i b$. S využitím (1.56) máme $a^i bb.a^{-j} = a^j bba^{-i}$. Pretože $b.b = b^2 = 1$ má posledná rovnica tvar $a^{i-j} = a^{j-i}$, čiže $a^{2(i-j)} = 1$. Pri riešení 1. prípadu sme však ukázali, že je to možné len ak $i = j$.

Veta 1.11. *Nech* $\mathbb{D}_n = \langle a, b \mid a^n = 1 = b^2,\ ba = a^{n-1}b \rangle$ *je Diedorva grupa nepárneho rádu* n, $n \geq 3$. *Nech permutácia* $\delta : \mathbb{D}_n \to \mathbb{D}_n$ *je definovaná predpisom* (1.54). *Definujme permutácie* $\delta_i = \delta^i$ *pre* $i = 1, 2, \ldots, m$. *Potom blokový kód dĺžky* m *s kontrolným znakom nad grupou* $\mathbb{D}_n$ *objavuje jednoduché chyby a susedné zámeny.*

Dôkaz. Podľa vety (1.7) stačí ukázať, že pre $x \neq y$

$$x * \delta_{i+1} \circ \delta_i^{-1}(y) \neq y * \delta_{i+1} \circ \delta_i^{-1}(x)$$

Keby pre nejaké $x \neq y$ v poslednej vzťahu platila rovnosť, dosadením za $\delta_i = \delta^i$, $\delta_{i+1} = \delta^{i+1}$ by sme dostali

$$\begin{aligned} x * \delta^{i+1} \circ \delta^{-i}(y) &= y * \delta^{i+1} \circ \delta^i(x) \\ x * \delta(y) &= y * \delta(x) \end{aligned}$$

čo by bolo v spore s vlastnosťami permutácie δ. □

Poznámka 1.2. Definíciu(1.54) možno zovšeobecniť nasledovne: Definujme $\delta : \mathbb{D}_n \to \mathbb{D}_n$ predpisom

$$\delta(a^i) = a^{c-i+d} \quad \text{a} \quad \delta(a^i b) = a^{i-c+d} b \quad \forall i = 1, 2, \ldots, n-1 \tag{1.57}$$

Potom definícia (1.54) je špeciálnym prípadom (1.57) pre $c = d = \dfrac{n-1}{2}$.

Príklad 1.14. Diederova grupa $\mathbb{D}_5 = \langle a, b \mid a^5 = 1 = b^2,\ ba = a^4 b \rangle$. Prvky grupy $\mathbb{D}_5$ možno priradiť dekadickým znakom nasledovne:

1	a	a^2	a^3	a^4	b	ab	a^2b	a^3b	a^4b
0	1	2	3	4	5	6	7	8	9

Pre grupovú operáciu $i * j$ bud platiť nasledovná schéma

$i * j$	$0 \leq j \leq 4$	$5 \leq j \leq 9$
$0 \leq i \leq 4$	$(i+j) \mod 5$	$5 + [(i+j) \mod 5]$
$5 \leq i \leq 9$	$5 + [(i-j) \mod 5]$	$(i-j) \mod 5$

z ktorej dostaneme tabuľku pre operáciu $*$

	j									
$*$	0	1	2	3	4	5	6	7	8	9
i 0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

1.9 Všeobecná teória kódov opravujúcich t jednoduchých chýb

Majme abecedu $A = \{a_1, a_2, \ldots, a_r\}$ s r znakmi. V tejto časti budeme skúmať blokové kódy dĺžky n, t.j. podmnožiny typu $\mathcal{K} \subset A^n$ z hľadiska všeobecných možnosti objavenia a opravy t jednoduchých chýb.

Podľa definície 1.5 je Hammingova vzdialenosť $d(\mathbf{a}, \mathbf{b})$ dvoch slov $\mathbf{a},\ \mathbf{b} \in A^n$ rovná počtu miest, na ktorých majú slová $\mathbf{a}$, $\mathbf{b}$ rôzne znaky. Minimálna vzdialenosť $\Delta\mathcal{K}$ je podľa definície 1.5 rovná minimu zo vzdialeností všetkých dvojíc kódu $\mathcal{K}$. Kód $\mathcal{K}$ objavuje t-násobné jednoduché chyby, ak pri zmene ľubovoľných t znakov slova $\mathbf{c}$ vznikne nekódové slovo. Ak teda prijmeme nekódové slovo, hovoríme, že sme objavili chybu.

Maximum vzdialeností dvoch slov z A^n môže byť n – to v prípade, keď príslušné slová nemajú ani na jednom mieste rovnaký znak.

Veta 1.12. *Hammingova vzdialenosť je metrikou na A^n, t.j. platí:*

$$d(\mathbf{a}, \mathbf{b}) \geq 0, \quad d(\mathbf{a}, \mathbf{b}) = 0 \iff \mathbf{a} = \mathbf{b} \tag{1.58}$$

$$d(\mathbf{a}, \mathbf{b}) = d(\mathbf{b}, \mathbf{a}) \tag{1.59}$$

$$d(\mathbf{a}, \mathbf{b}) \leq d(\mathbf{a}, \mathbf{c}) + d(\mathbf{c}, \mathbf{b}) \tag{1.60}$$

(A^n, d) je teda metrický priestor.

Dôkaz. Overenie vlastností metriky je jednoduché a prenechávame ho čitatľovi.

Definícia 1.9. Guľa $K_t(\mathbf{c})$ o strede $\mathbf{c} \in A^n$ a polomere t je množina

$$K_t(\mathbf{c}) = \{\mathbf{x} \mid \mathbf{x} \in A^n,\ d(\mathbf{x}, \mathbf{c}) \leq t\} \tag{1.61}$$

$K_t(\mathbf{c})$ je množina všetkých takých slov, ktoré vznikli zo slova $\mathbf{c}$ nanajvýš t jednoduchými chybami. Skúmajme, koľko prvkov obsahuje $K_t(\mathbf{c})$. Počet slov, ktoré sa líšia od $\mathbf{c} \in A^n$ práve na jednom mieste je rovný $n.(r-1) = \binom{n}{1}.(r-1)$, pretože na každom z n miest slova $\mathbf{c}$ zmenou pôvodného znaku na niektorý iný dostaneme $r-1$ rôznych slov, ktoré sa líšia od $\mathbf{c}$ len na jednom mieste (pripomeňme, že $|A| = r$). Počet slov, ktoré majú od slova $\mathbf{c}$ vzdialenosť práve 2 je $\binom{n}{2}.(r-1)^2$, pretože dvojicu zmenených znakov slova $\mathbf{c}$ možno vybrať $\binom{n}{2}$ spôsobmi a každú takúto dvojicu možno nahradiť $(r-1)^2$ spôsobmi znakmi rôznymi od pôvodných. Podobne sa ukáže, že počet slov, ktoré majú vzdialenosť od slova $\mathbf{c}$ rovnú i je $\binom{n}{i}.(r-1)^i$. Samotné slovo $\mathbf{c}$ je tiež prvkom gule $K_t(\mathbf{c})$ a prispieva k počtu jej prvkov číslom $1 = \binom{n}{0}.(r-1)^0$. Počet slov v $K_t(\mathbf{c})$ je teda

$$|K_t(\mathbf{c})| = \sum_{i=0}^{t} \binom{n}{i}.(r-1)^i \tag{1.62}$$

Počet prvkov gule $K_t(\mathbf{c})$ nezávisí na tom, aké slovo $\mathbf{c}$ sme vybrali za jej stred – všetky gule o rovnakom polomere t majú rovnkú mohutnosť (1.62).

Definícia 1.10. Hovoríme, že **kód** $\mathcal{K}$ **opravuje** t **jednoduchých chýb**, ak pre slovo $\mathbf{y}$, ktoré vzniklo z niektorého kódového slova nanajvýš t jednoduchými chybami, existuje jediné slovo $\mathbf{x}$ také, že $d(\mathbf{x}, \mathbf{y}) \leq t$.

Všimnime si, že ak $\mathbf{b} \in K_t(\mathbf{c}_1) \cap K_t(\mathbf{c}_2)$, potom slovo $\mathbf{b}$ mohlo vzniknúť nanavýš t jednoduchými chybami z oboch slov $\mathbf{c}_1$, $\mathbf{c}_2$. Ak má teda kód $\mathcal{K}$ opravovať t chýb, musí byť pre ľubovoľnú dvojicu $\mathbf{c}_1$, $\mathbf{c}_2$ rôznych kódových slov

$$K_t(\mathbf{c}_1) \cap K_t(\mathbf{c}_2) = \emptyset \tag{1.63}$$

Predpokladajme, že kód $\mathcal{K} \subseteq A^n$ opravuje t jednoduchých chýb. Keďže $|A^n| = r^n$, pre počet kódových slov $|\mathcal{K}|$ vzhľadom na (1.62) a (1.63) platí

$$\sum_{i=0}^{t} \binom{n}{i}.(r-1)^i\ .\ |\mathcal{K}| \leq r^n \tag{1.64}$$

Pri návrhu kódu opravujúceho t jednoduchých chýb sa snažíme čo najlepšie využiť priestor (A^n, d). Ideálne by bolo, keby sme dostali taký systém disjunktných gulí o polomere t, ktorý by pokrýval celú množinu A^n, t.j. keby v (1.64) platila rovnosť.

Definícia 1.11. Hovoríme, že **kód** $\mathcal{K} \subseteq A^n$ **je** t**-perfektný kód**, ak

$$\forall \mathbf{a},\ \mathbf{b} \in A^n, \quad \mathbf{a} \neq \mathbf{b} \quad K_t(\mathbf{a}) \cap K_t(\mathbf{b}) = \emptyset \tag{1.65}$$

$$\bigcup_{\mathbf{a} \in \mathcal{K}} K_t(\mathbf{a}) = A^n \tag{1.66}$$

Veta 1.13. *Kód $\mathcal{K}$ opravuje t-násobné chyby práve vtedy, keď*

$$\Delta(\mathcal{K}) \geq 2t + 1, \tag{1.67}$$

kde $\Delta(\mathcal{K}) = \min_{\mathbf{a},\mathbf{b} \in \mathcal{K}}\{d(\mathbf{a}, \mathbf{b})\}$.

Dôkaz. Nech platí (1.67). Keby existovali $\mathbf{a} \in \mathcal{K}$ $\mathbf{b} \in \mathcal{K}$ také, že $K_t(\mathbf{a}) \cap K_t(\mathbf{b}) \neq \emptyset$, vezmeme $\mathbf{c} \in K_t(\mathbf{a}) \cap K_t(\mathbf{b})$. Podľa trojuholníkovej nerovnosti platí

$$d(\mathbf{a}, \mathbf{b}) \leq \underbrace{d(\mathbf{a}, \mathbf{c})}_{\leq t} + \underbrace{d(\mathbf{c}, \mathbf{b})}_{\leq t} \leq 2t,$$

čo je v spore s prepokladom, že $\Delta\mathcal{K} \geq 2t + 1$.

Nech kód $\mathcal{K} \subseteq A^n$ opravuje t jednoduchých chýb. Potom pre ľubovoľné $\mathbf{a},\ \mathbf{b} \in \mathcal{K}$ je $K_t(\mathbf{a}) \cap K_t(\mathbf{b}) = \emptyset$. Keby $d(\mathbf{a}, \mathbf{b}) = s \leq 2t$, vytvorme postupnosť

$$\mathbf{a}_0, \mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s \tag{1.68}$$

tak, že položíme $\mathbf{a}_0 = \mathbf{a}$ a keď už máme definované $\mathbf{a}_i$, definujeme $\mathbf{a}_{i+1}$ nasledovne: Postupne prechádzame znakmi slova $\mathbf{a}_i$ a porovnávame ich so znakmi slova $\mathbf{b}$ na rovnakej pozícii. Keď poprvýkrát natrafíme na znak slova $\mathbf{a}_i$ na pozícii k, ktorý sa nezhoduje s rovnako položeným znakom slova $\mathbf{b}$, definujeme slovo $\mathbf{a}_{i+1}$ ako slovo $\mathbf{a}_i$ so zameneným znakom na pozícii k k-tym znakom slova $\mathbf{b}$. Postupnosť (1.68) predstavuje jeden z možných spôsobov, ako sa postupne pôsobením jednej jednoduchej chyby za druhou transformovalo slovo $\mathbf{a}$ na slovo $\mathbf{b}$.

Vidíme, že $a_s = b$, $d(\mathbf{a}, \mathbf{a}_i) = i$ a $d(\mathbf{a}_i, \mathbf{b}) = s - i$ pre $i = 1, 2, \ldots, s$. Preto $d(\mathbf{a}, \mathbf{a}_t) = t$ $\mathbf{a}_t \in K_t(\mathbf{a})$ a tiež $d(\mathbf{a}_t, \mathbf{b}) = s - t \leq 2t - t = t$, a teda $\mathbf{a}_t \in K_t(\mathbf{b})$, čo je v spore s predpokladom, že $K_t(\mathbf{a}) \cap K_t(\mathbf{b}) = \emptyset$. □

Príklad 1.15. Majme abecedu $A = \{a_1, a_2, \ldots, a_r\}$. Opakovací kód dĺžky k je blokový kód, ktorého kódové slová pozostávajú z k rovnakých znakov, t.j. $\mathcal{K} = \{a_1a_1 \ldots a_1,\ a_2a_2 \ldots a_2,\ \ldots\ a_ra_r \ldots a_r\}$. Minimálna vzdialenosť opakovacieho kódu dĺžky k je $\Delta\mathcal{K} = k$ a takýto kód opravuje t-násobné chyby pre $t < k/2$. Pre k nepárne, t.j. $k = 2t + 1$ je opakovací kód t-perfektný.

Príklad 1.16. Kód s kontrolou parity (pozri príklad 1.6) má minimálnu vzdialenosť 2 a preto neopravuje ani jednu jednoduchú chybu.

Príklad 1.17. Kód dvojrozmernej kontroly parity. Je to binárny kód, pri ktorom informačné znaky zapíšeme do matice typu (p, q). Potom ku každému riadku pridáme jeden symbol kontroly parity riadku a ku každému stĺpcu pridáme jeden symbol kontroly parity stĺpca – oba kontrolné znaky tak, aby sme dosiahli párnu paritu riadkov i stĺpcov a nakoniec pridáme znak „kontroly kontrol“ tak, aby aj parita výslednej matice bola párna. Tento kód opraví jednu jednoduchú chybu – takáto chyba zmení paritu práve jedného riadku i a práve jedného stĺpca j potom chybný znak je na mieste (i, j). Príklad kódového slova pre $p = 3$, $q = 7$:

101	1 ← kontrola parity riadku
000	0
001	1
010	1
111	1
111	1
000	0
kontroly parity stĺpcov → 110	0 ← celková kontrola parity

Majme kód $\mathcal{K}$, ktorý opravuje t chýb. Ak sme už prijali nejaké slovo $\mathbf{a}$ (či už s chybami, alebo bez nich), potrebujeme predpis, ako zo slova $\mathbf{a}$ dostať pôvodné vyslené slovo bez chýb.

Definícia 1.12. Dekódovanie kódu $\mathcal{K}$ je ľubovoľné zobrazenie δ , ktorého definičný obor $\mathcal{D}(\delta)$ je podmnožinou množiny A^n všetkých slov dĺžky n, obsahuje $\mathcal{K}$ a pre ľubovoľné $\mathbf{a} \in \mathcal{K}$ je $\delta(\mathbf{a}) = \mathbf{a}$. Ak $\mathcal{D}(\delta) = a^n$, hovoríme, že dekódovanie δ je **úplné**, inak hovoríme, že dekódovanie δ je **čiastočné**.

Pri definícii pojmu„dekódovanie“ prichádza trochu k nasledujúcemu terminologickému problému: Ak je zobrazenie K kódovanie, potom dekódovaním by malo byť inverzné zobrazenie K^{-1}. Lenže pri problematike kódov opravujúcich t chýb nezávisí ani tak na tvare kódovania K, ako na vlastnostiach množiny kódových slov. Preto do úvah o objavovaní a opravovaní chýb ani konkrétny tvar kódovania nezahŕňame. Ak chceme pri prenose slov s chybami zistiť, aký znak x bol zakódovaný a poslaný, keď sme prijali slovo $\mathbf{a}$, identifikujeme znak x ako $x = K^{-1} \circ \delta(\mathbf{a})$. Určiť tvar inverzného zobrazenia k vzájomne jednoznačnému zobrazeniu K nebýva ťažké, problémom však býva určiť zobrazenie δ. Aby sme ani do ďalších úvah nemuseli zavádzať konkrétny tvar kódovania K, dohodneme sa, že pod dekódovaním budeme rozumieť zobrazenie δ podľa definície 1.12 tak, ako ho používa i väčšina literatúry o kódovaní.

U niektorých kódov môžeme rozlíšiť jednotlivé znaky na informačné a kontrolné. Kontrolné znaky sú úplne určené informačnými znakmi. Napríklad dvanásťmiestne medzinárodné číslo vagónu má prvých jedenásť znakov informačných a jeden – posledný dvanásty znak kontrolný. Podobne je to s ISBN číslom knihy, či EAN kódom tovar atď. Kód s kontrolou parity dĺžky 8 má 7 informačných znakov a jeden kontrolný znak.

Ak vieme, ako sú definované jednotlivé položky resp. znaky kódových slov, nie je problém rozdeliť znaky na informačné a kontrolné. Ako to však urobiť pre kód $\mathcal{K} \subseteq A^n$, keď poznáme len to, ako vyzerá množina kódových slov? Odpoveď dáva nasledujúca definícia:

Definícia 1.13. Nech $\mathcal{K} \subseteq A^n$ je blokový kód dĺžky n. Hovoríme, že **kód $\mathcal{K}$ má k informačných a $n-k$ kontrolných znakov**, ak existuje vzájomne jednoznačné zobrazenie $\phi : A^k \leftrightarrow \mathcal{K}$. Zobrazenie ϕ nazveme **kódovanie informačných znakov**.

Príklad 1.18. Opakovací kód dĺžky 5 s abecedou $A = \{0,1,2,3,4,5,6,7,8,9\}$ má jeden informačný znak a 4 znaky kontrolné, pretože zobrazenie ϕ definované

$$\begin{array}{lllll} \phi(0) = 00000 & \phi(1) = 11111 & \phi(2) = 22222 & \phi(3) = 33333 & \phi(4) = 44444 \\ \phi(5) = 55555 & \phi(6) = 66666 & \phi(7) = 77777 & \phi(8) = 88888 & \phi(9) = 99999 \end{array}$$

je vzájomne jednoznačné zobrazenie $\phi : A^1 \leftrightarrow \mathcal{K}$.

Príklad 1.19. Zdvojovací kód dĺžky $2n$ má n informačných a n kontrolných znakov. Kódovanie informačných znakov $\phi : A^n \leftrightarrow \mathcal{K}$ definujeme predpisom

$$\phi(a_1 a_2 \dots a_n) = a_1 a_1 a_2 a_2 \dots a_n a_n$$

Príklad 1.20. Kód dva z piatich (pozri príklad 1.5) vôbec nemá oddelené informačné a kontrolné znaky. Počet kódových slov tohoto kódu je 10 – nie je mocninou čísla 2, a preto nemôže existovať vzájomne jednoznačné zobracenie množiny $\{0,1\}$ na množinu kódových slov mohutnosti 10.

V mnohých príkladoch sme videli, že kontrolná číslica bola posledným znakom kódového slova. Podobne by sme si priali, aby aj pri kódoch s k informačnými a $n-k$ kontrolnými znakmi najprv v kódovom slove vystupovali informačné a až potom kontrolné znaky. Takéto kódovanie informačných znakov nazývame systematické. Presne tento pojem určuje nasledujúca definícia:

Definícia 1.14. Blokový kód $\mathcal{K}$ je **systematický**, ak pre každé slovo $a_1 a_2 \dots a_k \in A^k$ existuje práve jedno kódové slovo $\mathbf{a} \in \mathcal{K}$ také, že

$$\mathbf{a} = a_1 a_2 \dots a_k, a_{k+1} \dots a_n$$

Príklad 1.21. Opakovací kód je systematický s $k = 1$. Kód s kontrolou parity dĺžky 8 je systematický s $k = 7$. Kód medzinárodného čísla vozňa je systematický s $k = 11$.

Príklad 1.22. Zdvojovací kód s dĺžkou $2n$ väčšou ako 2 nie je systematický.

Veta 1.14. *Nech $\mathcal{K}$ je systematický kód s k informačnými a $n-k$ kontrolnými znakmi. Potom pre minimálnu vzdialenosť $\Delta\mathcal{K}$ platí*

$$\Delta\mathcal{K} \leq n-k+1 \tag{1.69}$$

Dôkaz. Zvoľme dve slová $\mathbf{a} = a_1a_2\ldots a_{k-1}a_k \in A^k$, $\overline{\mathbf{a}} = a_1a_2\ldots a_{k-1}\overline{a}_k \in A^k$ líšiace sa len v poslednom k-tom znaku. Pretože kód $\mathcal{K}$ je systematický, ku každému z takýchto slov existuje práve jedno slovo $\mathbf{b}$ resp. $\overline{\mathbf{b}}$ kódu $\mathcal{K}$, také, že $\mathbf{a}$ je prefixom $\mathbf{b}$, resp. $\overline{\mathbf{a}}$ je prefixom $\overline{\mathbf{b}}$.

$$\mathbf{b} = a_1a_2\ldots a_{k-1}a_ka_{k+1}\ldots a_n$$
$$\overline{\mathbf{b}} = a_1a_2\ldots a_{k-1}\overline{a}_k\overline{a}_{k+1}\ldots\overline{a}_n$$

Keďže sa slová $\mathbf{b}$, $\overline{\mathbf{b}}$ zhodujú na $k-1$ miestach môžu sa nezhodovať najviac na $n-(k-1)=n-k+1$ miestach. Je $d(\mathbf{b},\overline{\mathbf{b}}) \leq n-k+1$ a teda $\Delta\mathcal{K} \leq n-k+1$. □

Dôsledok Kód $\mathcal{K}$ s k informačnými a $n-k$ kontrolnými znakmi môže opravovať najviac $\left[\frac{n-k}{2}\right]$ chýb (kde $[x]$ je celá časť čísla x).

Príklad 1.23. Pre zdvojovací kód dĺžky $n=2t$ je $k=t$, $n-k=t$, ale minimálna vzdialenosť tohoto kódu je 2, čo je pre veľké t hlboko pod odhadom (1.69), ktorý pre náš prípad dáva $\Delta\mathcal{K} \leq 2t-t+1 = t+1$.

Definícia 1.15. Nech $\mathcal{K}$ je kód s k informačnými a $n-k$ kontrolnými znakmi. Pomer

$$R = \frac{k}{n} \tag{1.70}$$

nazveme **informačný pomer**.

Pri navrhovaní samoopravných kódov sa snažíme zabezpečiť sa proti čo najväčšiemu počtu chýb čo vedie k zvyšovaniu počtu kontrolných znakov. Druhou prirodzenou požiadavkou je dosiahnuť čo najväčší informačný pomer, čo je v rozpore so zvyšovaním počtu kontrolných znakov. Naviac na príklade 1.23 vidíme, že nie každé zvyšovanie počtu kontrolných znakov musí viesť k zväčšovaniu minimálnej vzdialenosti kódu.

1.10 Pripomenutie niektorých algebraických štruktúr

Grupa je množina G spolu s binárnou operáciou . priraďujúcou každým dvom prvkom $a \in G$, $b \in G$ prvok $a.b$ (krátko len ab) tak, že platí:

(i) $\forall a,b \in G\ ab \in G$

(ii) $\forall a,b,c \in G$ (ab)c=a(bc) – asociatívny zákon

(iii) $\exists 1 \in G\ \forall a \in G \quad 1a = a1 = a$ – existenica neutrálneho prvku

(iv) $\forall a \in G\ \exists a^{-1} \in G \quad aa^{-1} = a^{-1} = 1$ – pre každý prvok grupy existuje inverzný prvok.

Grupa G ke komutatívna, ak platí $\forall a,b \in G ab = ba$. V tomto prípade sa zvykne grupová operácia zapisovať aditívne, t.j. $a+b$ namiesto $a.b$ a neutrálny prvok sa pri aditívnom zápise označuje ako 0. Inverzný prvok k prvku a sa v komutatívnom prípade nazýva opačný prvok a označuje sa $-a$.

Teleso je množina T obsahujúca (okrem iných prvkov) prvky 0 a 1 spolu s operáciami + a . takými, že platí:

(i) Množina T spolu s operáciou + je komutatívna grupa s neutrálnym prvkom 0.

(ii) Množina $T - \{0\}$ spolu s operáciou . je komutatívna grupa s neutrálnym prvkom 1.

(iii) $\forall a, b, c \in G \quad a(b + c) = ab + ac$ – platí distributívny zákon

Vlastnosti telesa si možno lepšei uvedomíme, ak (i), (ii), (iii) definície rozpíšeme na jednotlivé konkrétne podmienky, ktoré musí teleso spĺňať:
Teleso je množina T taká, že prvky $0 \in T$ a $1 \in T$ spolu s operáciami + a . takými, že platí:

(T1) $\forall a, b \in T \; a + b \in T, \; ab \in T.$

(T2) $\forall a, b, c \in T \; a + (b + c) = (a + b) + c, \; a(bc) = (ab)c$ – platia asociatívne zákony.

(T3) $\forall a, b \in T \; a + b = b + a, \; ab = ba$ – platia komutatívne zákony.

(T4) $\forall a, b, c \in T \; a(b + c) = ab + ac$ – platia distributívne zákony.

(T5) $\forall a \in T \; a+0 = a, \; a.1 = a$ – 0 je neutrálny prvok vzhľadom k operácii „+“, 1 je neutrálny prvok vzhľadom k operácii „.“.

(T6) $\forall a \in T \; \exists(-a) \in T \; a + (-a) = 0$ – ku každému prvku T existuje opačný prvok.

(T7) $\forall a \in T, \; a \neq 0 \; \exists a^{-1} \in T \; a.a^{-1} = 1$ – ku každému prvku T rôznemu od nuly existuje inverzný prvok.

Komutatívny okruh s jednotkou je množina R taká, že $0 \in R$, $1 \in R$ spolu s operáciami + a ., v ktorej platia (T1) až (T6).

Príklad 1.24. Množina celých čísel spolu s operáciami + a . je komutatívnym okruhom s jednotkou.

Faktorový okruh modulo p. Majme množinu $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$. Na množine $\mathbb{Z}_p$ definujeme operácie $\oplus$, $\otimes$ nasledujúcim spôsobom

$$a \oplus b = (a + b) \mod p \qquad a \otimes b = (ab) \mod p, \tag{1.71}$$

kde $n \mod p$ je zvyšok po celočíselnom delení čísla n číslom p. Ľahko sa dá ukázať, že pre ľubovoľné prirodzené číslo $p > 1$ je $\mathbb{Z}_p$ spolu s operáciami $\oplus$, $\otimes$ komutatívnym okruhom s jednotkou, t.j. spĺňa požiadavky (T1) až (T6).

Na $\mathbb{Z}_p$ sa môžeme pozerať i s nasledujúceho hľadiska. **Triedou modulo** p nazveme podmnožinu okruhu $\mathbb{Z}$ celých čísel takú, že rozdiel jej ľubovoľných dvoch prvkov je deliteľný číslom p. Triedu označujeme jej ľubovoľným reprezentanto v hranatej zátvorke. Môžeme teda triedu modulo p obsahujúcu celé číslo a definovať nasledovne:

$$[a] = \{a + pk \mid k = 0, +1, -1, +2, -2, \ldots\} \tag{1.72}$$

Ľahko sa ukáže, že ak dve triedy $[a]$, $[b]$ majú aspoň jeden spoločný prvok, potom $[a] = [b]$. Ďalej je vidieť, že všetky triedy okruhu $\mathbb{Z}$ modulo p sú $[0], [1], \ldots, [p-1]$.
Ak dva prvky a, a' majú tú istú triedu modulo p, budeme písať

$$a \equiv a' \mod p \tag{1.73}$$

a hovoriť, že prvky a, a' sú **kongruentné modulo p.**
Označme $\mathbb{Z}_p$ množinu všetkých tried modulo p, t.j. $\mathbb{Z}_p = \{[0], [1], \ldots, [p-1]\}$. Na $\mathbb{Z}_p$ definujeme operácie + a . predpisom:

$$[a] + [b] = [a + b] \qquad [a].[b] = [ab] \tag{1.74}$$

Ukazuje sa, že takto definované operácie sčítania a násobenia sú korektne definované – t.j. nezávisí na výbere reprezentanta triedy. Množina $\mathbb{Z}_p$ s takto definovanými operáciami + a . je znovu

komutatívnym okruhom s jednotkou a nazývame ju **faktorovým okruhom okruhu celých čísel** $\mathbb{Z}$ **modulo** p alebo len faktorovým okruhom modulo p.
Obiddve reprezentácie $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$, $\mathbb{Z}_p = \{[0], [1], \ldots, [p-1]\}$ sú ekvivalentné, pre ľubovoľné $a, b, c \in \{0, 1, 2, \ldots, p-1\}$ je $a \oplus b = c$ práve vtedy, keď $[a] + [b] = [c]$, $a \otimes b = c$ práve vtedy, keď $[a].[b] = [c]$; rozdiel je len v označení prvkov a operácií. Budeme preto používať jednoduchšiu prvú reprezentáciu, kde naviac budeme namiesto $\oplus$ a $\otimes$ používať $+$ a . keď nedôjde k nedorozumeniu.

Príklad 1.25. Okruh $\mathbb{Z}_6$ bude mať nasledujúce tabuľky pre operácie sčítania a násobenia:

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	5

.	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	2	2	1

Podľa vyššie uvedených tabuliek je $5.5 = 1$, t.j. inverzným prvkom prvku 5 je prvok 5. Prvky 2, 3, 4 vôbec nemajú inverzný prvok. Podmienka (T7) v $\mathbb{Z}_6$ nie je splnená – $\mathbb{Z}_6$ nie je telesom.

Pre potreby kódovania budú výhodné také faktorové okruhy $\mathbb{Z}$, ktoré sú telesami. Je teraz pred nami otázka, kedy je $\mathbb{Z}_p$ telesom. Odpoveď dáva nasledujúca veta.

Veta 1.15. *Faktorový okruh $\mathbb{Z}_p$ je telesom práve vtedy, keď p je prvočíslo.*

Lineárne priestory nad telesom T**.** Nech T je teleso. Lineárnym priestorom nad telesom T je množina $\mathcal{L}$ spolu s binárnou operáciou $+$ (sčítanie) a skalárnou operáciou . (skalárne násobenie) takými, že platí

(L1) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L}$ a $\forall t \in T \quad u + v \in \mathcal{L}$, $t.\mathbf{u} \in \mathcal{L}$.

(L2) $\forall \mathbf{u}, \mathbf{v}, \mathbf{w} \in \mathcal{L} \qquad \mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.

(L3) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L} \qquad \mathbf{u} + \mathbf{b} = \mathbf{b} + \mathbf{u}$.

(L4) $\exists \mathbf{o} \in \mathcal{L}$ také, že $\forall \mathbf{u} \in \mathcal{L} \quad \mathbf{u} + \mathbf{o} = \mathbf{u}$

(L5) $\forall \mathbf{u} \in \mathcal{L} \quad \exists(-\mathbf{u}) \in \mathcal{L}$ také, že $\mathbf{u} + (-\mathbf{u}) = \mathbf{o}$

(L6) $\forall \mathbf{u}, \mathbf{v} \in \mathcal{L}$ a $\forall t \in T \quad t.(\mathbf{u} + \mathbf{v}) = t.\mathbf{u} + t.\mathbf{v}$

(L7) $\forall \mathbf{u} \in \mathcal{L}$ a $\forall s, t \in T \quad (s.t)\mathbf{u} = s.(t.\mathbf{u})$

(L8) $\forall \mathbf{u} \in \mathcal{L}$ a $\forall s, t \in T \quad (s + t)\mathbf{u} = s.\mathbf{u} + t.\mathbf{u}$

(L9) $\forall \mathbf{u} \in \mathcal{L} \quad 1.u = u$.

Požiadavky (L1) až (L5) sú ekvivalentné s požiadavku, aby $(\mathcal{L}, +)$ bola komutatívna grupa s neutrálnym prvkom $\mathbf{o}$. Pre lineárne priestory sa používa synonymum **vektorové priestory**, ich prvky sa volajú **vektory**.

Vektory $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ sa nazývajú **lineárne nezávislé**, ak zo vzťahu $\sum_{i=1}^{n} t_i \mathbf{u}_i = \mathbf{o}$ vyplýva $t_i = 0$ pre $i = 1, 2, \ldots, n$. Hovoríme, že lineárny priestor $\mathcal{L}$ je **konečne dimenzionálny**, ak existuje také prirodzené číslo k, že každá $k + 1$ prvková množina vektorov z $\mathcal{L}$ je lineárne závislá. V konečne dimenzionálnom priestore majú všetky maximálne nezávislé množiny vektorov rovnakú mohutnosť. Mohutnosť n maximálnej lineárne nezávislej podmnožiny $\mathcal{L}$ sa nazýva **dimenzia** lineárneho priestoru

$\mathcal{Z}$ – v tomto prípade hovoríme, že priestor je n-dimenzionálny. **Báza** konečne dimenzionálneho lineárneho priestoru je ľubovoľná maximálne nezávislá množina jeho vektorov.

Lineárny priestor T^n je priestor n-prvkových postupností typu $\mathbf{u} = u_1u_2\dots u_n$, kde $u_i \in T$ a kde je sčítanie a skalárne násobenie definované nasledovne:
Nech $\mathbf{u} = u_1u_2\dots u_n$, $\mathbf{v} = v_1v_2\dots v_n$, $t \in T$. Potom

$$\mathbf{u} + \mathbf{v} = (u_1 + v_1)(u_2 + v_2)\dots(u_n + v_n) \qquad t.\mathbf{u} = (tu_1)(tu_2)\dots(tu_n). \tag{1.75}$$

Skalárny súčin vektorov u, v je definovaný nasledovne

$$\mathbf{u} * \mathbf{v} = u_1v_1 + u_2v_2 + \dots + u_nv_n \tag{1.76}$$

Hovoríme, že vektory **u**, **v** sú **ortogonálne**, ak $\mathbf{u} * \mathbf{v} = 0$.

Dôležitosť priestoru T^n vzplýva z nasledujúcej vety:

Veta 1.16. *Každý n-dimenzionálny vektorový priestor nad telesom T je izomorfný s priestorom T^n.*

V teórii lineárnych kódov sa vychádza z toho, že na kódovej abecede sú dané opernácie + a . , s ktorými je táto abeceda konečným telesom. Potom sa na množinu všetkých n-znakových slov v kódovej abcede možno pozerať ako na n-dimenzionálny lineárny priestor. Videli sme už, že faktorové okruhy $\mathbb{Z}_p$ pre p prvočíslo sú telesami. Okrem toho existujú konečné telesá o p^n prvkoch (pozor, nie sú to však okruhy $\mathbb{Z}_{p^n}$). Mohutnosť kódovej abecedy je pre takéto úvahy obmedzená na čísla typu p^n, kde p je prvočíslo, t.j. 2,3,4,5,7,8,9,11,13,16,17 ... , ale nemôže byť 6,10,12,14,15 lebo tieto čísla nie sú mocninami prvočísiel. Tieto obmedzenia však nie sú tragické, pretože najdôležitejšou kódovou abecedou je binárna abeceda, pre abecedy s väčším počtom znakov použijeme najbližšie teleso s väčším počtom prvkov s tým, že niektoré z nich nevyužijeme.

1.11 Lineárne kódy

V tejto časti budeme predpokladať, že kódová abeceda $A = \{a_1, a_2, \dots, a_p\}$ má p prvkov, kde p je pvočíslo. Ďalej predpokladáme, že na abecede A sú definované operácie sčítania + a súčinu . , s ktorými A vytvára teleso. Množinu A^n n-znakových slov pokladáme za lineárny priestor s obvykle definovaným súčtom a skalárnym násobkom vektorov.

Definícia 1.16. Kód $\mathcal{K}$ sa nazýva **lineárny** (n, k)**-kód**, ak je podpriestorom dimenzie k lineárneho priestoru A^n, t.j. ak $\dim(\mathcal{K}) = k$, a pre ľubovoľné $\mathbf{a}, \mathbf{b} \in \mathcal{K}$ a ľubovoľné $c \in A$ je

$$\mathbf{a} + \mathbf{b} \in \mathcal{K}, \qquad c.\mathbf{a} \in \mathcal{K} \tag{1.77}$$

Lineárny (n, k)-kód ako k-dimenzionálny podpriestor priestoru A^n musí mať k-prvkovú bázu $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$. Potom každé kódové slovo $\mathbf{a} \in A^n$ má jednoznačné vyjadrenie

$$\mathbf{a} = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \dots + a_k\mathbf{b}_k, \tag{1.78}$$

kde $a_1, a_2, \dots, a_n$ sú súradnice vektora **a** v báze **B**. Ak $|A| = p$, potom na mieste každého a_i môže stáť p rôznych čísel, z čoho vyplýva, že existuje p^k rôznych k-tic $a_1, a_2, \dots, a_k$, dosadením ktorých do (1.78) dostaneme p^k rôznych kódových slov kódu $\mathcal{K}$. Lineárny (n, k)-kód má teda p^k slov.

Všimnime si priradenie $\phi : A^k \to A^n$ definované

$$\forall(a_1a_2\dots a_n) \quad \phi(a_1a_2\dots a_n) = a_1\mathbf{b}_1 + a_2\mathbf{b}_2 + \dots + a_k\mathbf{b}_k.$$

Zobrazenie ϕ je vzájomne jednoznačné zobrazenie $A^k \leftrightarrow \mathcal{K}$ a teda podľa definície 1.13 má lineárny (n, k)-kód $\mathcal{K}$ k informačných a $n - k$ kontrolných znakov. Zobrazenie ϕ je kódovanie informačných znakov.

Definícia 1.17. Nech $\mathcal{K}$ je lineárny (n,k)-kód, nech $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$ je ľubovoľná báza kódu $\mathcal{K}$. Nech $\mathbf{b}_i = (b_{i1}\ b_{i2} \ldots b_{in})^T$ pre $i = 1, 2, \ldots, k$. Potom maticu

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \ldots \\ \mathbf{b}_k^T \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ b_{k1} & b_{k2} & \ldots & b_{kn} \end{bmatrix} \tag{1.79}$$

typu $(k \times n)$ sa nazýva **generujúca matica kódu $\mathcal{K}$.**

Poznámka 1.3. Podľa definície 1.17 je generujúcou maticou kódu $\mathcal{K}$ každá matica, ktorej

a) každý riadok je kódovým slovom

b) riadky sú lineárne nezávislé, takže hodnosť matice $\mathbf{G}$ je rovná k.

c) každé kódové slovo je lineárnou kombináciou riadkov matice

Ak teda z matice $\mathbf{G}$ vytvoríme ekvivalentnými riadkovými úpravami ekvivalentnú maticu $\mathbf{G}'$, potom aj matica $\mathbf{G}'$ je generujúcou maticou kódu $\mathcal{K}$.

Poznámka 1.4. Často bude výhodné využívať maticové zápisy, v ktorých vektory budú vystupovať ako jednoriadkové alebo jednostĺpcové matice. Dohodneme sa, že **slová - t.j. vektory $\mathbf{a} \in A^n$ budeme v maticových zápisoch vždy považovať za stĺpcové matice**, t.j. ak $\mathbf{a} = a_1 a_2 \ldots a_k$ sa vyskytne v maticovom zápise, budeme predpokladať, že

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \ldots \\ a_k \end{bmatrix} .$$

Ak budeme potrebovať vektor $\mathbf{a}$ v tvare jednoriadkovej matice, zapíšeme ho ako $\mathbf{a}^T$, t.j.

$$\mathbf{a}^T = \begin{bmatrix} a_1 & a_2 & \ldots & a_k \end{bmatrix} .$$

Skalárny súčin dvoch vektorov $\mathbf{u},\ \mathbf{v} \in A^n$ môžeme považovať za súčin matíc a zapísať ako $\mathbf{u}^T.\mathbf{v}$.

Poznámka 1.5. Nech má lineárny (n,k)-kód pre bázu $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k\}$ generujúcu maticu (1.79). Ak má slovo $\mathbf{a} = a_1 a_2 \ldots a_n$ súradnice $u_1, u_2, \ldots, u_k$ v báze $\mathbf{B}$, potom

$$\mathbf{a}^T = u_1\mathbf{b}_1^T + u_2\mathbf{b}_2^T + \cdots + u_k\mathbf{b}_k^T = \begin{bmatrix} u_1 & u_2 & \ldots & u_k \end{bmatrix} . \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \ldots \\ \mathbf{b}_k^T \end{bmatrix}$$

skrátene:

$$\begin{bmatrix} a_1 & a_2 & \ldots & a_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \ldots & u_k \end{bmatrix} . \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1n} \\ b_{21} & b_{22} & \ldots & b_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ b_{k1} & b_{k2} & \ldots & b_{kn} \end{bmatrix}$$

čo v maticovom tvare môžeme napísať

$$\mathbf{a}^T = \mathbf{u}^T.\mathbf{G} \tag{1.80}$$

Príklad 1.26. Príklady lineárnych kódov.
a) Binárny kód dĺžky 4 s kontrolou parity – lineárny $(4,3)$-kód:
$\mathcal{K} \subset A^4, \quad A = \{0,1\}:$ 0000, 00011, 0101, 0110
1001, 10010, 1100, 1111
Báza: $B = \{0011, 0101, 1001\}$.

$$\text{Generujúca matica} \quad \mathbf{G} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

b) Ternárny opakovací kód dĺžky 5 – lineárny $(5,1)$-kód :
$\mathcal{K} \subset A^5, \quad A = \{0,1,2\}:$ 00000, 11111, 22222
Báza: $\{11111\}$.

$$\text{Generujúca matica} \quad \mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

c) Binárny zdvojovací kód dĺžky 6 – lineárny $(6,3)$-kód :
$\mathcal{K} \subset A^6, \quad A = \{0,1\}:$ 000000, 000011, 001100, 001111
110000, 110011, 111100, 111111
Báza: $\{000011, 001100, 110000\}$.

$$\text{Generujúca matica} \quad \mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

d) Dekadický kód dĺžky n s kontrolnou číslicou modulo 10 nie je lineárnym kódom, lebo neexistuje konečné teleso s počtom prvkov 10.

Definícia 1.18. Hovoríme, že dva blokové kódy $\mathcal{K}$, $\mathcal{K}'$ dĺžky n sú **ekvivalentné**, ak existuje permutácia π množiny $\{1,2,\ldots,n\}$ taká, že platí

$$a_1 a_2 \ldots a_n \in \mathcal{K} \quad \text{práve vtedy, keď} \quad a_{\pi[1]} a_{\pi[2]} \ldots a_{\pi[n]} \in \mathcal{K} \tag{1.81}$$

Podľa definície 1.14 je blokový kód $\mathcal{K}$ s k informačnými a $n-k$ kontrolnými znakmi systematický, ak ku každému $a_1 a_2 \ldots a_k \in A^k$ existuje práve jedno slovo $\mathbf{a} \in A^n$ s prefixom $a_1 a_2 \ldots a_k \in A^k$. Ako sme už ukázali, lineárny (n,k)-kód je kódom s k informačnými a $n-k$ kontrolnými znakmi, avšak nemusí byť systematický. Zdvojovací kód je dĺžky $n = 2k$ lineárny kód, ktorý nie je systematický. Stačí však zmeniť poradie znakov v slove $a_1 a_2, \ldots a_n$ dať najprv znaky na nepárnych miestach a potom znaky na párnych miestach a takto získaný nový kód je už systematický. Toto sa dá urobiť s každým lineárnym (n,k)-kódom.

Veta 1.17. *Lineárny (n,k)-kód $\mathcal{K}$ je systematický práve vtedy, keď k nemu existuje generujúca matica* $\mathbf{G}$ *typu:*

$$\mathbf{G} = \left[\, \mathbf{E} \,\middle|\, \mathbf{B} \,\right] = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & b_{11} & b_{12} & \ldots & h_{1n-k} \\ 0 & 1 & 0 & \ldots & 0 & b_{21} & b_{22} & \ldots & b_{2n-k} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & b_{k1} & h_{k2} & \ldots & h_{kn-k} \end{bmatrix} . \tag{1.82}$$

Dôkaz. Nech (1.82) je generujúcou maticou pre kód $\mathcal{K}$. Nech $\mathbf{u} = u_1, u_2, \dots u_k$ sú súradnice slova $\mathbf{a} = a_1 a_2 \dots a_n \in \mathcal{K}$ v báze, ktorú tvoria riadky generujúcej matice $\mathbf{G}$. Potom podľa poznámky 1.5 je $\mathbf{a}^T = \mathbf{b}^T.\mathbf{G}$. Špeciálne pre $\mathbf{u} = a_1 a_2 \dots a_k$ je

$$\mathbf{u}^T.\mathbf{G} = \left[\begin{array}{cccc} a_1 & a_2 & \dots & a_k \end{array}\right] . \left[\begin{array}{cccccccc} 1 & 0 & 0 & \dots & 0 & b_{11} & b_{12} & \dots & b_{1n-k} \\ 0 & 1 & 0 & \dots & 0 & b_{21} & b_{22} & \dots & b_{2n-k} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & b_{k1} & b_{k2} & \dots & b_{kn-k} \end{array}\right] =$$

$$= \left[\begin{array}{ccccccc} a_1 & a_2 & \dots & a_k & v_{k+1} & \dots & v_n \end{array}\right],$$

kde v_{k+i} je jednoznačne určené vzťahom:

$$v_{k+i} = \left[\begin{array}{cccc} a_1 & a_2 & \dots & a_k \end{array}\right] . \left[\begin{array}{c} b_{1i} \\ b_{2i} \\ \dots \\ b_{ki} \end{array}\right] .$$

Pre každé $a_1 a_2 \dots a_k \in A^k$ existuje práve jedno slovo kódu $\mathcal{K}$ s prefixom $a_1 a_2 \dots a_k$. Kód $\mathcal{K}$ je teda systematický.

Nech je kód $\mathcal{K}$ systematický. Ak sú prvé k stĺpce generujúcej matice $\mathbf{G}$ lineárne nezávislé, riadkovými ekvivalentnými úpravami ju môžeme previesť na ekvivalentnú maticu $\mathbf{G}'$ v tvare $\mathbf{G}' = \left[\begin{array}{c|c} \mathbf{E} & \mathbf{B} \end{array}\right]$, ktorá je tiež generujúcou maticou kódu $\mathcal{K}$.

Nech teda prvé k stĺpce generujúcej matice $\mathbf{G}$ systematického kódu $\mathcal{K}$ nie sú lineárne závislé. Potom ju môžeme ekvivalentnými riadkovými úpravami transformovať na ekvivalentný tvar

$$\mathbf{G}' = \left[\begin{array}{ccccc|ccccc} d_{11} & d_{12} & d_{13} & \dots & d_{1k} & d_{1(k+1)} & d_{1(k+2)} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2k} & d_{2(k+1)} & d_{2(k+2)} & \dots & d_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ d_{(k-1)1} & d_{(k-1)2} & d_{(k-1)3} & \dots & d_{(k-1)k} & d_{(k-1)(k+1)} & d_{(k-1)(k+2)} & \dots & d_{(k-1)n} \\ 0 & 0 & 0 & \dots & 0 & d_{k(k+1)} & d_{k(k+2)} & \dots & d_{kn} \end{array}\right] \quad (1.83)$$

Matica $\mathbf{G}'$ má hodnosť k, pretože je ekvivalentná s maticou $\mathbf{G}$, ktorá mala k lineárne nezávislých riadkov. Pre $\mathbf{u}, \mathbf{v} \in A^k$ také, že $\mathbf{u} \neq \mathbf{v}$ je $\mathbf{u}^T.\mathbf{G}' \neq \mathbf{v}^T.\mathbf{G}'$. Všimnime si, že prvých k súradníc vektora $\mathbf{u}^T.\mathbf{G}$ nezávisí na k-tej súradnici vektora $\mathbf{u}$, z čoho vyplýva, že existuje niekoľko kódových slov kódu $\mathcal{K}$ s rovnakým prefixom a teda kód $\mathcal{K}$ nie je systematický. Z predpokladu, že prvé k stĺpce generujúcej matice sú závislé, sme dostali spor. □

Dôsledok. Lineárny (n, k)-kód $\mathcal{K}$ je systematický práve vtedy, keď jeho ľubovoľná generujúca matica $\mathbf{G}$ má prvé k stĺpce lineárne nezávislé.

Veta 1.18. *Každý lineárny (n, k)-kód $\mathcal{K}$ je ekvivalentný so systematickým lineárnym kódom.*

Dôkaz. Nech $\mathbf{G}$ je generujúca matica (n, k) kódu $\mathcal{K}$. Matica $\mathbf{G}$ má k lineárne nezávislých riadkov a preto musí mať (apoň jednu) k-ticu lineárne nezávislých stĺpcov. Ak prvých k stĺpcov matice $\mathbf{G}$ je linárne nezávislých, podľa dôsledku vety 1.17 je kód $\mathcal{K}$ systematický.

Ak sú prvé k stĺpce lineárne závislé, urobíme takú permutáciu π stĺpcov, aby prvé k stĺpce boli lineárne nezávislé. Potom už príslušný kód $\mathcal{K}'$, ktorý dostaneme rovnakou permutáciou π znakov kódu $\mathcal{K}$ bude systematický. □

Existuje i iný spôsob charakterizácie lineárneho (n, k)-kódu a to tak, že vlastnosti kódových slov charakterizujem sústavou lineárnych rovníc, ktoré musia všetky kódové slová spĺňovať. Tak napríklad binárny kód dĺžky n s kontrolou parity charakterizujeme rovnicou:

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

Zdvojovací kód dĺžky $n = 2k$ charakterizujeme sústavou rovníc:

$$\begin{aligned} x_1 - x_2 &= 0 \\ x_3 - x_4 &= 0 \\ &\dots \\ x_{n-1} - x_n &= 0 \end{aligned}$$

Sústava rovníc pre opakovací kód dĺžky n bude:

$$\begin{aligned} x_1 - x_2 &= 0 \\ x_1 - x_3 &= 0 \\ &\dots \\ x_1 - x_n &= 0 \end{aligned}$$

Definícia 1.19. Kontrolná matica lineárneho kódu $\mathcal{K}$ je taká matica $\mathbf{H}$ prvkov kódovej abecedy A, pre ktorú platí: Slovo $\mathbf{v} = v_1 v_2 \dots v_n$ je kódové práve vtedy, keď

$$\mathbf{H}.\mathbf{v} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \dots & \dots & \dots & \dots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{bmatrix} . \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} = \mathbf{o} \tag{1.84}$$

Stručnejšie: $\mathbf{v} \in \mathcal{K}$ práve vtedy, keď $\mathbf{H}.\mathbf{v} = \mathbf{o}$.

Majme lineárny (n, k)-kód $\mathcal{K}$ s generujúcou maticou

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \dots \\ \mathbf{b}_k^T \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \dots & \dots & \dots & \dots \\ b_{k1} & b_{k2} & \dots & b_{kn} \end{bmatrix} \tag{1.85}$$

typu $(k \times n)$. Aká má byť kontrolná matica kódu $\mathcal{K}$ t.j matica $\mathbf{H}$ taká, že $\mathbf{H}.\mathbf{u} = \mathbf{o}$ práve vtedy, keď $\mathbf{u} \in \mathcal{K}$? Prvé, čo o matici $\mathbf{H}$ vieme povedať, je, že má mať n stĺpcov (už len preto, aby $\mathbf{H}.\mathbf{u}$ bolo definované pre $\mathbf{u} \in A^n$). Množina všetkých $\mathbf{u} \in A^n$ takých, že $\mathbf{H}.\mathbf{u} = \mathbf{o}$ je podpriestor priestoru A^n dimenzie rovnej $n - \dim(\mathbf{H}) = \dim(\mathcal{K}) = k$, odkiaľ $\dim(\mathbf{H}) = n - k$. Stačí teda hľadať maticu $\mathbf{H}$ ako maticu typu $((n-k) \times n)$ s $n-k$ lineárne nezávislými riadkami. Nech $\mathbf{h}^T$ je riadok matice $\mathbf{H}$. Potom pre každé kódové slovo $\mathbf{u} \in \mathcal{K}$ musí byť

$$\mathbf{u}^T.\mathbf{h} = u_1 h_1 + u_2 h_2 + \dots + u_n h_n = 0. \tag{1.86}$$

Mohli by sme teda zostaviť sústavu $p^k = |\mathcal{K}|$ lineárnych rovníc typu (1.86), kde by $\mathbf{u}$ prebiehalo všetky kódové slová kódu $\mathcal{K}$. Takýto systém lineárnych rovníc by však obsahoval príliš veľa lineárne závislých rovníc. Stačí totiž, aby (1.86) platilo pre všetky prvky bázy podpriestoru $\mathcal{K}$, potom bude (1.86) platiť aj pre všetky prvky podpriestoru $\mathcal{K}$. Pre $\mathbf{h}$ možno zostaviť túto sústavu lineárnych rovníc:

$$\left.\begin{aligned} \mathbf{b}_1^T.\mathbf{h} &= 0 \\ \mathbf{b}_2^T.\mathbf{h} &= 0 \\ &\dots \\ \mathbf{b}^T.\mathbf{h} &= 0 \end{aligned}\right\} \tag{1.87}$$

čo sa dá v maticovom tvare zapísať

$$\mathbf{G}.\mathbf{h} = \mathbf{o}. \tag{1.88}$$

Keďže dimenzia matice $\mathbf{G}$ je k, množina všetkých riešení sústavy (1.88) je podpriestor dimenzie $(n-k)$ a preto možno nájsť $(n-k)$ lineárne nezávislých riešení sústavy (1.88) $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n-k}$, ktoré budú riadkami hľadanej kontrolnej matice $\mathbf{H}$, t.j

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \ldots \\ \mathbf{h}_{n-k}^T \end{bmatrix} \tag{1.89}$$

Všimnime si, že

$$\mathbf{G}.\mathbf{H}^T = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \ldots \\ \mathbf{b}_k^T \end{bmatrix}_{k\times n} . \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_{n-k} \end{bmatrix}_{n\times(n-k)} = \begin{bmatrix} 0 & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{k\times(n-k)} \tag{1.90}$$

Majme maticu $\mathbf{H}$ typu $((n-k)\times n)$ dimenzie $\dim(\mathbf{H}) = (n-k)$ pre ktorú platí $\mathbf{G}.\mathbf{H}^T = \mathbf{O}_{k\times(n-k)}$, kde $\mathbf{O}_{k\times(n-k)}$ je nulová matica typu $(k\times(n-k))$. Označme $\mathcal{N} \subseteq A^n$ priestor všetkých riešení rovnice $\mathbf{Hu} = \mathbf{o}$. Keďže pre všetky prvky bázy kódu $\mathbf{K}$ platí $\mathbf{H}.\mathbf{b}_i = \mathbf{o}$, $i = 1, 2, \ldots k$, platí aj pre ľubovoľné kódové slovo $\mathbf{u} \in \mathcal{K}$, $\mathbf{u} = \sum_{i=1}^{k} u_i \mathbf{b}_i$:

$$\mathbf{H}.\mathbf{u} = \mathbf{H}.\sum_{i=1}^{k} u_i\mathbf{b}_i = \sum_{i=1}^{k} \mathbf{H}.(u_i\mathbf{b}_i) = \sum_{i=1}^{k} u_i(\mathbf{H}.\mathbf{b}_i) = \sum_{i=1}^{k} u_i.\mathbf{o} = \mathbf{o}$$

Máme teda $\mathcal{K} \subseteq \mathcal{N}$. Pretože $\dim(\mathbf{H}) = (n-k)$, je $\dim(\mathcal{N})$ rovná $n - \dim(\mathbf{H}) = k$. Keďže $\mathcal{K} \subseteq \mathcal{N}$ je báza $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k$ bázou priestoru $\mathcal{N}$, a teda $\mathcal{K} = \mathcal{N}$.

Práve dokázané skutočnosti môžeme sformulovať do nasledujúcej vety.

Veta 1.19. *Nech $\mathcal{K}$ je lineárny (n,k)-kód s generujúcou maticou $\mathbf{G}$ typu $(k\times n)$. Potom matica $\mathbf{H}$ typu $((n-k)\times n)$ je kontrolnou maticou kódu $\mathcal{K}$ práve vtedy, keď*

$$\dim(\mathbf{H}) = (n-k) \quad a \quad \mathbf{G}.\mathbf{H}^T = \mathbf{O}_{k\times(n-k)}, \tag{1.91}$$

kde $\mathbf{O}_{k\times(n-k)}$ je nulová matica typu $(k\times(n-k))$.

Pre systematické kódy je situácie jednoduchšia, ako hovorí nasledujúca veta.

Veta 1.20. *Lineárny (n,k)-kód $\mathcal{K}$ s generujúcou maticou $\mathbf{G} = \left[\begin{array}{c|c} \mathbf{E}_{k\times k} & \mathbf{B} \end{array}\right]$ má kontrolnú maticu $\mathbf{H} = \left[\begin{array}{c|c} -\mathbf{B}^T & \mathbf{E}_{(n-k)\times(n-k)} \end{array}\right]$.*

Dôkaz. Označme $m = n-k$. Potom môžeme matice $\mathbf{G}$, $\mathbf{H}$ rozpísať nasledovne:

$$\mathbf{G} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \ldots \\ \mathbf{b}_p^T \\ \ldots \\ \mathbf{b}_k^T \end{bmatrix} = \left[\begin{array}{ccccccc|ccccc} 1 & 0 & \ldots & 0 & \ldots & 0 & b_{11} & b_{12} & \ldots & b_{1q} & \ldots & b_{1m} \\ 0 & 1 & \ldots & 0 & \ldots & 0 & b_{21} & b_{22} & \ldots & b_{1q} & \ldots & b_{2m} \\ \ldots & & & & & & & & & & & \\ 0 & 0 & \ldots & 1 & \ldots & 0 & b_{p1} & b_{p2} & \ldots & b_{pq} & \ldots & b_{pm} \\ \ldots & & & & & & & & & & & \\ 0 & 0 & \ldots & 0 & \ldots & 1 & b_{k1} & b_{k2} & \ldots & b_{kq} & \ldots & b_{km} \end{array}\right] \tag{1.92}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \ldots \\ \mathbf{h}_q^T \\ \ldots \\ \mathbf{h}_m^T \end{bmatrix} = \left[\begin{array}{ccccccc|ccccc} -b_{11} & -b_{21} & \ldots & -b_{p1} & \ldots & -b_{k1} & 1 & 0 & \ldots & 0 & \ldots & 0 \\ -b_{12} & -b_{22} & \ldots & -b_{p2} & \ldots & -b_{k2} & 0 & 1 & \ldots & 0 & \ldots & 0 \\ \ldots & & & & & & & & & & & \\ -b_{1q} & -b_{2q} & \ldots & -b_{pq} & \ldots & -b_{kq} & 0 & 0 & \ldots & 1 & \ldots & 0 \\ \ldots & & & & & & & & & & & \\ -b_{1m} & -b_{2m} & \ldots & -b_{pm} & \ldots & -b_{km} & 0 & 0 & \ldots & 0 & \ldots & 1 \end{array}\right] \tag{1.93}$$

Pre $\mathbf{b}_p$, $\mathbf{h}_q$ platí

$$\begin{array}{rcl} \mathbf{b}_p^T & = & [\ 0 \quad 0 \quad \dots \quad 1 \quad \dots \quad 0 \quad b_{p1} \quad b_{p2} \quad \dots \quad b_{pq} \quad \dots \quad b_{pm}\ ] \\ \mathbf{h}_q^T & = & [\ -b_{1q} \quad -b_{2q} \quad \dots \quad -b_{pq} \quad \dots \quad -b_{kq} \quad 0 \quad 0 \quad \dots \quad 1 \quad \dots \quad 0\ ] \end{array}$$

a preto je $\mathbf{b}_p^T.\mathbf{h}_q = (-b_{pq} + b_{pq}) = 0$ pre každé $p,\ q \in \{1, 2, \dots, n\}$, z čoho

$$\mathbf{G}.\mathbf{H}^T = \mathbf{O}_{k\times(n-k)}. \tag{1.94}$$

Keďže matica $\mathbf{H}$ s $m = n - k$ riadkami obsahuje jednotkovú podmaticu $\mathbf{E}_{(n-k)\times(n-k)}$, je $\dim(H) = n - k$. Podľa vety je 1.19 je $\mathbf{H}$ kontrolnou maticou kódu $\mathcal{K}$.

Definícia 1.20. Nech $\mathcal{K} \subseteq A^n$ je lineárny (n, k) kód. Duálny kód $\mathcal{K}^\perp$ kódu $\mathcal{K}$ definujeme ako

$$\mathcal{K}^\perp = \{\mathbf{v} \mid \mathbf{a}.\mathbf{v} = 0\ \forall \mathbf{a} \in \mathcal{K}\}. \tag{1.95}$$

Veta 1.21. *Nech $\mathcal{K} \subseteq A^n$ je lineárny (n, k)-kód s generujúcou maticou $\mathbf{G}$ a kontrolnou maticou $\mathbf{H}$. Potom duálny kód $\mathcal{K}^\perp$ kódu $\mathcal{K}$ je lineárny $(n, n - k)$-kód s generujúcou maticou $\mathbf{H}$ a kontrolnou maticou $\mathbf{G}$.*

Dôkaz. Platí $\mathbf{v} \in \mathcal{K}^\perp$ práve vtedy, keď

$$\mathbf{G}.\mathbf{v} = \mathbf{o}. \tag{1.96}$$

Pretože $\mathcal{K}^\perp$ je množinou riešení rovnice (1.96) a $\dim(\mathbf{G}) = k$, je $\mathcal{K}^\perp$ $(n - k)$-dimenzionálnym podpriestorom A^n – t.j. lineárnym $(n, (n - k))$-kódom s kontrolnou maticou $\mathbf{G}$.

Pretože $\mathbf{H}.\mathbf{G}^T = \left((\mathbf{G}^T)^T.\mathbf{H}^T\right)^T = \left(\mathbf{G}.\mathbf{H}^T\right)^T = \mathbf{O}_{k\times(n-k)}^T = \mathbf{O}_{(n-k)\times k}$, je každý riadok matice $\mathbf{H}$ ortogonálny s podpriestorom $\mathcal{K}$ a teda kódovým slovo kódu $\mathcal{K}^\perp$. Pretože $\dim(\mathbf{H}) = (n - k)$, generujú riadky matice $\mathbf{H}$ celý priestor $\mathcal{K}^\perp$, t.j. matica $\mathbf{H}$ je generujúcou maticou kódu $\mathcal{K}^\perp$.

Príklad 1.27. Duálny kód binárneho opakovacieho kódu $\mathcal{K}$ dĺžky 5 je kód obsahujúci všetky binárne slová $v_1v_2 \dots v_n$ také, že

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

Kód $\mathcal{K}^\perp$ je kód kontroly paritou.

Príklad 1.28. Duálny kód k binárnemu zdvojovaciemu kódu $\mathcal{K}$ je samotný kód $\mathcal{K}$, teda $\mathcal{K}^\perp = \mathcal{K}$. Ten má totiž generujúcu maticu

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{1.97}$$

Ľahko sa presvedčíme, že $\mathbf{G}.\mathbf{G}^T = \mathbf{O}_{3\times3}$ – generujúca matica binárneho zdvojovacieho kódu $\mathcal{K}$ je zároveň i jeho kontrolnou maticou.

1.12 Lineárne kódy a objavovanie chýb

V časti 1.6 v definícii 1.5 sme všeobecne definovali, čo to znamená, že kód objavuje niekoľkonásobné chyby – totiž kód $\mathcal{K}$ objavuje t-násobné jednoduché chyby, ak pri zmene ľubovoľných t znakov ľubovoľného kódového slova $\mathbf{u}$ vznikne nekódové slovo.

Teória lineárnych kódov nám umožňuje podrobnejšie modelovať mechanizmus vzniku niekoľkonásobnej chyby a to tak, ako keby sa k vyslanému slovu $\mathbf{v} = v_1v_2 \dots v_n$ behom prenosu pripočítalo slovo $\mathbf{e} = e_1e_2 \dots e_n$. Potom namiesto slova $\mathbf{v}$ prijmeme slovo $\mathbf{w} = w_1w_2 \dots w_n$ pre ktoré platí $\mathbf{w} = \mathbf{v} + \mathbf{e}$. Slovo $\mathbf{e}$ nazývame **chybové slovo**.

Definícia 1.21. Hovoríme, že lineárny kód $\mathcal{K}$ **objavuje chybové slovo e**, ak pre každé kódové slovo $\mathbf{v}$ je slovo $\mathbf{v}+\mathbf{e}$ nekódovým slovom.

Definícia 1.22. Hammingova váha $\|\mathbf{a}\|$ slova $\mathbf{a} \in A^n$ je počet nenulových znakov slova $\mathbf{a}$.

Každý binárny lineárny kód obsahuje buď len slová párnej váhy, alebo má rovnaký počet slov párnej a nepárnej váhy. Skutočne, ak existuje kódové slovo $\mathbf{v}$ nepárnej váhy, fixujme ho a definujme zobrazenie $f : \mathcal{K} \to \mathcal{K}$ predpisom

$$f(\mathbf{w}) = \mathbf{w} + \mathbf{v}.$$

Je ihneď vidieť, že f je vzájomne jednoznačné zobrazenie $\mathcal{K}$ na $\mathcal{K}$ priraďujúce každému slovu párnej váhy slovo nepárnej váhy a naopak. Z toho už vyplýva, že počet slov párnej váhy je rovný počtu slov nepárnej váhy.

Všimnime si, že lineárny kód objavuje t-násobné chyby práve vtedy, keď objavuje všetky chybové slová Hammingovej váhy menšej alebo rovnej t.

Pre objavovanie a opravovanie chýb má podstatný význam minimálna vzdialenosť $\Delta(\mathcal{K})$ blokového kódu $\mathcal{K}$, ktorá bola v definíci 1.5 definovaná ako minimum z Hammingových vzdialeností všetkých dvojíc slov kódu $\mathcal{K}$. Ak totiž $d = \Delta(\mathcal{K})$, kód $\mathcal{K}$ objavuje všetky $(d-1)$-násobné chyby a opravuje všetky t-násobné chyby pre $t < \dfrac{d}{2}$ (pozri vetu 1.13).

Pre lineárny kód sa $\Delta(\mathcal{K})$ určí ešte jednoduchšie.

Veta 1.22. *Pre lineárny kód $\mathcal{K}$ je minimálna vzdialenosť kódu $\Delta(\mathcal{K})$ rovná minimum z Hammingových váh všetkých nenulových slov kódu $\mathcal{K}$, t.j.*

$$\Delta(\mathcal{K}) = \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}} \{\|\mathbf{u}\|\} \tag{1.98}$$

Dôkaz. 1. Majme $\mathbf{u},\ \mathbf{v} \in \mathcal{K}$ také, že $d(\mathbf{u},\mathbf{v}) = \Delta(\mathcal{K})$. Nech $\mathbf{w} = \mathbf{u} - \mathbf{v}$. Slovo $\mathbf{w}$ má práve toľko nenulových znakov, v koľkých znakoch sa líšia slová $\mathbf{u},\ \mathbf{v}$, preto je

$$\min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}} \{\|\mathbf{u}\|\} \leq \|\mathbf{w}\| = d(\mathbf{u},\mathbf{v}) = \Delta(\mathcal{K}) \tag{1.99}$$

2. Vezmime $\mathbf{w} \in \mathcal{K}$ také, že $\|w\| = \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}}\{\|\mathbf{u}\|\}$. Potom

$$\Delta(\mathcal{K}) \leq d(\mathbf{o},\mathbf{v}) = \|\mathbf{w}\| \leq \min_{\mathbf{u}\in\mathcal{K},\mathbf{u}\neq\mathbf{o}} \{\|\mathbf{u}\|\} \tag{1.100}$$

Vzťahy (1.99) a (1.100) už dávajú tvrdenie vety. □

Definícia 1.23. Nech $\mathbf{H}$ je kontrolná matica lineárneho kódu $\mathcal{K}$, nech $\mathbf{v} = v_1v_2\ldots v_n \in A^n$ je ľubovoľné slovo abecedy A dĺžky n. **Syndrom slova v** je slovo $\mathbf{s} = s_1s_2\ldots s_n$, pre ktoré platí

$$\mathbf{H}.\begin{bmatrix} v_1 \\ v_2 \\ \cdots \\ v_n \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \cdots \\ s_n \end{bmatrix}, \qquad \text{skrátene} \quad \mathbf{H}.\mathbf{v} = \mathbf{s}. \tag{1.101}$$

Ak teda prijmeme slovo $\mathbf{w}$, vypočítame jeho syndrom $\mathbf{s} = \mathbf{H}\mathbf{w}$ a ak $\mathbf{s} \neq \mathbf{o}$, vieme, že došlo k chybe. Naviac vieme, že syndrom prijatého slova $\mathbf{w} = \mathbf{v} + \mathbf{e}$ je rovnaký, ako syndrom chybového slova $\mathbf{e}$. Je totiž

$$\mathbf{H}\mathbf{w} = \mathbf{H}(\mathbf{v}+\mathbf{e}) = \mathbf{H}\mathbf{v} + \mathbf{H}\mathbf{e} = \mathbf{o} + \mathbf{H}\mathbf{e} = \mathbf{H}\mathbf{e}.$$

Keďže kód $\mathcal{K}$ je podpriestor práve všetkých riešení rovnice $\mathbf{H} = \mathbf{o}$, rovnica $\mathbf{H}\mathbf{e} = \mathbf{s}$ má množinu všetkých riešení v tvare $\mathbf{e} + \mathbf{a}$, kde $\mathbf{k} \in \mathcal{K}$. Túto množinu budeme v ďalšom označovať ako $\mathbf{e} + \mathcal{K}$.

Veta 1.23. *Nech $\mathcal{K}$ je lineárny kód s kontrolnou maticou $\mathbf{H}$. Nech d je minimum z počtu lineárne závislých stĺpcov kontrolnej matice $\mathbf{H}$. Potom pre minimálnu vzdialenosť $\Delta(\mathcal{K})$ kódu $\mathcal{K}$ platí*

$$d = \Delta(\mathcal{K}). \tag{1.102}$$

Dôkaz. Podľa vety 1.22 je $\Delta(\mathcal{K})$ rovné minimálnej váhe nenulového kódového slova. Nech d je minimum počtu lineárne závislých stĺpcov kontrolnej matice $\mathbf{H}$.
Označme $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n$ stĺpce kontrolnej matice $\mathbf{H}$, t.j.

$$\mathbf{H} = \left[\begin{array}{cccc} \mathbf{c}_1 & \mathbf{c}_2 & \ldots & \mathbf{c}_n \end{array} \right]$$

Nech $\mathbf{u} \in \mathcal{K}$ je nenulové slovo s najmenšou Hammingovou váhou $\|\mathbf{u}\| = t$. Slovo $\mathbf{u}$ má miestach $i_1, i_2, \ldots, i_t$ znaky $u_{i_1}, u_{i_2}, \ldots, u_{i_t}$ a na ostatných miestach znak 0, t.j.

$$\mathbf{u}^T = \left[\begin{array}{ccccccccccccccccc} 0 & 0 & \ldots & 0 & u_{i_1} & 0 & \ldots & 0 & u_{i_2} & 0 & \ldots & \ldots & 0 & u_{i_t} & 0 & \ldots & 0 & 0 \end{array} \right]$$

Pretožu $\mathbf{u}$ je kódové slovo, je $\mathbf{Hv} = \mathbf{o}$, t.j.

$$\mathbf{Hu} = \sum_{i=1}^{n} u_i.\mathbf{c}_i = u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2}, \ldots, u_{i_t}\mathbf{u}_{i_t} = \mathbf{o} \tag{1.103}$$

Pretože všetky koeficienty u_{i_j} sú nenulové, sú stĺpce $\mathbf{c}_{i_1}, \mathbf{c}_{i_2}, \ldots, \mathbf{c}_{i_t}$ lineárne závislé a preto

$$d \leq \Delta(\mathcal{K}). \tag{1.104}$$

Majme d lineárne závislých stĺpcov $\mathbf{c}_{i_1}, \mathbf{c}_{i_2}, \ldots, \mathbf{c}_{i_d}$. Potom existujú čísla $u_{i_1}, u_{i_2}, \ldots, u_{i_d}$ také, že $u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2}, \ldots, u_{i_d}\mathbf{u}_{i_d} = \mathbf{o}$, z ktorých aspoň jedno je nenulové. Definume slovo $\mathbf{u}$ také, že na miestach $i_1, i_2, \ldots, i_d$ bude mať znaky $u_{i_1}, u_{i_2}, \ldots, u_{i_d}$ a na ostatných miestach znak 0, t.j.

$$\mathbf{u}^T = \left[\begin{array}{ccccccccccccccccc} 0 & 0 & \ldots & 0 & u_{i_1} & 0 & \ldots & 0 & u_{i_2} & 0 & \ldots & \ldots & 0 & u_{i_t} & 0 & \ldots & 0 & 0 \end{array} \right]$$

Potom

$$\mathbf{Hu} = \sum_{i=1}^{n} u_i.\mathbf{c}_i = u_{i_1}\mathbf{c}_{i_1} + u_{i_2}\mathbf{c}_{i_2}, \ldots, u_{i_t}\mathbf{u}_{i_d} = \mathbf{o} \tag{1.105}$$

a teda $\mathbf{u}$ je nenulovým kódovým slovom, pre ktorého Hammingovu váhu platí $\|\mathbf{u}\| \leq d$ a teda

$$\Delta(\mathcal{K}) \leq d. \tag{1.106}$$

Posledná nerovnosť spolu s (1.104) už dáva požadované tvrdenie vety. □

Veta 1.24. *Lineárny kód objavuje t-násobné chyby práve vtedy, keď každých t stĺpcov kontrolnej matice je lineárne nezávislých.*

Dôkaz. Označme $d = \Delta(\mathcal{K})$. Podľa predchádzajúcej vety 1.23 existuje v kontrolnej matici $\mathbf{H}$ kódu $\mathcal{K}$ d lineárne závislých stĺpcov, a pre každé $t < d$ je ľubovoľných t stĺpcov matice $\mathbf{H}$ lineárne nezávislých.

Ak kód $\mathcal{K}$ objavuje t-násobné chyby, potom musí byť $t < d$ a podľa vety 1.23 je každých t stĺpcov matice $\mathbf{H}$ lineárne nezávislých.

Ak je každých t stĺpcov matice $\mathbf{H}$ lineárne nezávislých, potom podľa vety 1.23 je $t < d$ a preto kód $\mathcal{K}$ objavuje t chýb.

1.13 Štandardné dekódovanie

V predchádzajúcej časti sme ukázali, ako určíme najväčší počet jednoduchých chýb, ktoré je kód $\mathcal{K}$ schopný objaviť a akým spôsobom zistíme, či nastalo niekoľko jednoduchých chýb (pochopiteľne za predpokladu, že ich je najviac t). Ak už prijmeme nekódové slovo, chceme mu priradiť kódové slovo, ktorého pokazením toto slovo pravdepodobne vzniklo (za predpokladov, že počet chýb neprekročil hodnotu t). Na to slúži dekódovanie δ definované v časti 1.9 v definícii 1.12 ako funkcia, ktorá ma za definičný obor A^n alebo jeho časť obsahujúcu kód $\mathcal{K}$ a ktorá každému slovu zo svojho definičného oboru priraďuje kódové slovo, pričom je δ na $\mathcal{K}$ identitou – kódovému slovu $\mathbf{a}$ priraďuje $\delta(\mathbf{a}) = \mathbf{a}$.

Ak bolo vyslané slovo $\mathbf{v}$ a došlo k chybe vyjadrenej slovom $\mathbf{e}$, prijmeme slovo $\mathbf{e} + \mathbf{v}$. Ak $\delta(\mathbf{e} + \mathbf{v}) = \mathbf{v}$, dekódovali sme správne.

Definícia 1.24. Hovoríme, že **lineárny kód $\mathcal{K}$ pri dekódovaní δ opravuje chybové slovo e, ak pre všetky $\mathbf{v} \in \mathcal{K}$ platí:**

$$\delta(\mathbf{u} + \mathbf{v}) = \mathbf{v}. \tag{1.107}$$

Definícia 1.25. Nech $\mathcal{K} \subseteq A^n$ je lineárny kód s kódovou abecedou A. Pre každé slovo $\mathbf{e} \in A^n$ definujeme

$$\mathbf{e} + \mathcal{K} = \{\mathbf{e} + \mathbf{v} \mid \mathbf{v} \in \mathcal{K}\} \tag{1.108}$$

Množina $\mathbf{e} + \mathcal{K}$ sa volá **trieda slova e podľa kódu** $\mathcal{K}$.

Veta 1.25. *Nech $\mathcal{K} \subseteq A^n$ je lineárny (n, k)-kód s kódovou abecedou A, $|A| = p$. Pre ľubovoľné slová $\mathbf{e}$, $\mathbf{e}' \in A^n$ platí*

(i) Ak $\mathbf{e} - \mathbf{e}'$ je kódové slovo, potom $\mathbf{e} + \mathcal{K} = \mathbf{e}' + \mathcal{K}$.

(ii) Ak $\mathbf{e} - \mathbf{e}'$ nie je kódové slovo, potom $\mathbf{e} + \mathcal{K}$, $\mathbf{e}' + \mathcal{K}$ sú disjunktné.

(iii) Počet slov každej triedy je rovný počtu kódových slov, t.j. $|\mathbf{e} + \mathcal{K}| = |\mathcal{K}| = p^k$ a počet všetkých tried je p^{n-k}.

Dôkaz. (i) Nech $(\mathbf{e} - \mathbf{e}') \in \mathcal{K}$, nech $\mathbf{v} \in \mathcal{K}$ a teda $(\mathbf{e} + \mathbf{v}) \in (\mathbf{e} + \mathcal{K})$. Položme $\mathbf{u} = \mathbf{v} + (\mathbf{e} - \mathbf{e}')$. Pretože $\mathcal{K}$ je lineárny priestor a $(\mathbf{e} - \mathbf{e}') \in \mathcal{K}$, je aj $\mathbf{u} \in \mathcal{K}$ a teda $(\mathbf{e}' + \mathbf{u}) \in (\mathbf{e}' + \mathcal{K})$. Ale $\mathbf{e}' + \mathbf{u} = \mathbf{e}' + \mathbf{v} + (\mathbf{e} - \mathbf{e}') = \mathbf{e} + \mathbf{v}$. Preto je $(\mathbf{e} + \mathbf{v}) \in (\mathbf{e}' + \mathcal{K})$. Ukázali sme, že $(\mathbf{e} + \mathcal{K}) \subseteq (\mathbf{e}' + \mathcal{K})$. Analogicky sa ukáže aj opačná inklúzia, a teda $(\mathbf{e} + \mathcal{K}) = (\mathbf{e}' + \mathcal{K})$.

(ii) Nech $(\mathbf{e} - \mathbf{e}') \notin \mathcal{K}$. Keby existovalo $\mathbf{w} \in (\mathbf{e}+\mathcal{K}) \cap (\mathbf{e}'+\mathcal{K})$, museli by existovať slová $\mathbf{v}$, $\mathbf{v}' \in \mathcal{K}$ také, že

$$\mathbf{w} = \mathbf{e} + \mathbf{v}$$
$$\mathbf{w} = \mathbf{e}' + \mathbf{v}',$$

odkiaľ máme $\mathbf{e} + \mathbf{v} = \mathbf{e}' + \mathbf{v}'$ a ďalej $\mathbf{e} - \mathbf{e}' = \mathbf{v}' - \mathbf{v} \in \mathcal{K}$ lebo obe slová $\mathbf{v}$, $\mathbf{v}'$ boli prvkami linárneho priestoru $\mathcal{K}$. Z predpokladu, že $(\mathbf{e} + \mathcal{K}) \cap (\mathbf{e}' + \mathcal{K}) \neq \emptyset$ sme dostali $(\mathbf{e} - \mathbf{e}') \in \mathcal{K}$ – čo je spor.

(iii) V úvahách bezprostredne po definícii 1.16 (str. 25) sme ukázali, že lineárny (n, k)-kód s p-prvkovou abecedou má p^k prvkov. Chceme ukázať, že $|\mathbf{e} + \mathcal{K}| = |\mathcal{K}| = p^k$. Na to stačí ukázať, že ak $\mathbf{u}$, $\mathbf{w} \in \mathcal{K}$, $\mathbf{u} \neq \mathbf{b}$, potom $\mathbf{e} + \mathbf{u} \neq \mathbf{e} + \mathbf{w}$. Keby však $\mathbf{e} + \mathbf{u} = \mathbf{e} + \mathbf{w}$, potom by (po odčítaní $\mathbf{e}$ od oboch strán rovnice) $\mathbf{u} = \mathbf{w}$. Všetky triedy slov podľa kódu $\mathcal{K}$ majú rovnaký počet prvkov p^k. Keďže zjadnotenie všetkých tried slov podľa kódu $\mathcal{K}$ je A^n a $|A^n| = p^k$ je

$$\text{Počet všetkých rôznych tried podľa kódu } \mathcal{K} \ = \frac{|A^n|}{|\mathcal{K}|} = \frac{p^n}{p^k} = p^{n-k}.$$

□

Definícia 1.26. Štandardné dekódovanie. Definujeme úplné dekódovanie $\delta : A^n \to \mathcal{K}$ nasledovne: Z každej triedy podľa $\mathcal{K}$ vyberieme jedného reprezentanta triedy tak, aby jeho váha bola v danej triede minimálna. (Výber reprezentanta podľa kritéria minimálnej váhy nemusí byť jednoznačný – v tom prípade sa musíme rozhodnúť pre jedného s minimálnou váhou). Potom každé prijaté slovo $\mathbf{w} \in A^n$ dekódujeme ako $\mathbf{v} = \mathbf{w} - \mathbf{e}$, kde chybové slovo $\mathbf{e}$ je reprezentantom triedy slova $\mathbf{w}$, teda

$$\delta(\mathbf{w}) = \mathbf{w} - [\text{reprezentant triedy } (\mathbf{w} + \mathcal{K})]. \tag{1.109}$$

Príklad 1.29. Binárny $(4, 3)$-kód $\mathcal{K}$ celkovej parity má dve triedy

$$\begin{aligned} 0000 + \mathcal{K} &= \{0000 \quad 0011 \quad 0101 \quad 0110 \quad 1001 \quad 1010 \quad 1100 \quad 1111\} \\ 0001 + \mathcal{K} &= \{0001 \quad 0010 \quad 0100 \quad 0111 \quad 1000 \quad 1011 \quad 1101 \quad 1110\} \end{aligned}$$

Trieda $0000 + \mathcal{K}$ má jednoznačného reprezentanta – slovo 0000. Trieda $0001 + \mathcal{K}$ môže mať za reprezentanta ľubovoľné zo slov 0001, 0010, 0100, 1000. Podľa toho, ktoré z týchto slov vyberieme za reprezentantov, štandardné dekódovanie opraví jednu chybu, ktorá vznikne na prvom, resp. druhom, treťom alebo štvrtom mieste. Ak vznikne chyba na inom mieste, štandartné dekódovanie nedekóduje správne. Pre nás to nie je prekvapujúce zistenie, lebo vieme, že kód celkovej parity má minimálnu vzdialenosť rovnú 2, a preto nemôže opravovať ani všetky jednoduché chyby.

Veta 1.26. *Štandardné dekódovanie δ opravuje práve tie chybové slová, ktoré sú reprezentantmi tried, t.j.*

$$\delta(\mathbf{v} + \mathbf{e}) = \delta(\mathbf{v})$$

práve vtedy, keď $\mathbf{e}$ je reprezentantom niektorej triedy podľa kódu $\mathcal{K}$.

Dôkaz. Ak je $\mathbf{e}$ reprezentantom svojej triedy a $\mathbf{v} \in \mathcal{K}$, potom slovo $\mathbf{v} + \mathbf{e}$ padne do triedy $\mathbf{e} + \mathcal{K}$ a dekóduje sa ako $\delta(\mathbf{e} + \mathbf{v}) = \mathbf{e} + \mathbf{v} - \mathbf{e} = \mathbf{v}$ – podľa definície 1.24 dekódovanie δ opravuje chybové slovo $\mathbf{e}$.

Nech $\mathbf{e}'$ nie je reprezentantom svojej triedy, ktorá má za reprezentanta slovo $\mathbf{e} \neq \mathbf{e}'$. Je $(\mathbf{e} - \mathbf{e}') \in \mathcal{K}$. Nech $\mathbf{v} \in \mathcal{K}$, potom slovo $\mathbf{v} + \mathbf{e}'$ padne do triedy $\mathbf{e} + \mathcal{K}$ a dekóduje sa ako $\delta(\mathbf{v} + \mathbf{e}') = \mathbf{v} + \mathbf{e}' - \mathbf{e} \neq \mathbf{v}$. Ak $\mathbf{e}'$ nie je reprezentantom svojej triedy, štandardné dekódovanie neopravuje slovo $\mathbf{e}'$.

Veta 1.27. *Štandardné dekódovanie δ je optimálne v tom zmysle, že neexistuje dekódovanie δ^*, ktoré by opravovalo tie isté chybové slová ako δ a naviac ešte niektoré ďalšie.*

Dôkaz. Vezmime $\mathbf{e}' \in (\mathbf{e} + \mathcal{K})$ nech $\mathbf{e}$ je reprezentantom triedy $\mathbf{e} + \mathcal{K}$, nech $\mathbf{e} \neq \mathbf{e}'$. Slovo $\mathbf{v} = \mathbf{e}' - \mathbf{e}$ je kódové a nenulové. Ak vyšleme slovo $\mathbf{v}$ a vznikne chyba pôsobením chybového slova $\mathbf{e}$, prijmeme slovo $\mathbf{v} + \mathbf{e} = \mathbf{e}' - \mathbf{e} + \mathbf{e} = \mathbf{e}'$. Keďže δ opravuje všetky slová, ktoré sú reprezentantami tried je $\delta(\mathbf{v} + \mathbf{e}) = \delta(\mathbf{e}') = v$. Keďže δ^* opravuje všetky slová, ktoré opravuje δ, je aj $\delta^*(\mathbf{e}') = \mathbf{v}$.

Môže dekódovanie δ^* opravovať slovo $\mathbf{e}'$? Keby áno, potom by muselo byť $\delta^*(\mathbf{o} + \mathbf{e}') = \mathbf{o}$, čo je v spore s tým, že $\delta^*(\mathbf{e}') = \mathbf{v} \neq \mathbf{o}$. □

Veta 1.28. *Ak je $d = \Delta(\mathcal{K})$ minimálna vzdialenosť lineárneho kód $\mathcal{K}$, potom štandardné dekódovanie opraví všetky t-násobné chyby pre $t < \frac{d}{2}$.*

Dôkaz. Nech $\mathbf{e}$ je slovo váhy $t < \frac{d}{2}$. Nech $\mathbf{v} \in (\mathbf{e} + \mathcal{K})$, $\mathbf{v} \neq \mathbf{e}$, $\mathbf{v} = \mathbf{e} + \mathbf{u}$, $\mathbf{u} \in \mathcal{K}$. Je $\|\mathbf{u}\| \geq d$, $\|\mathbf{e}\| = t < \frac{d}{2}$. Preto počet nenulových znakov slova $\mathbf{v} = \mathbf{e} + \mathbf{u}$ je aspoň $d - t$ – t.j. $\|\mathbf{v}\| > d - t > t$. Preto je každé slovo $\mathbf{e}$ s Hammingovou váhou menšou než $\frac{d}{2}$ reprezentantom niektorej triedy slov podľa kódu $\mathcal{K}$. Keďže štandardné dekódovanie opravuje všetky chybové slová, ktoré sú reprezentantami

tried, opravuje všetky chybové slová s Hammingovou váhou menšou než $\frac{d}{2}$ čo je ekvivalentné s tým, že štandartné dekódovanie opraví všetky t-násobné chyby. □

Princípom štandartného dekódovania je určenie, v ktorej triede slov podľa kódu $\mathcal{K}$ sa dekódované slovo vyskytuje. Na to by príslušný dekódovací algoritmus musel prezrieť tzv. Slepianovu tabuľku všetkých slov dĺžky n abecedy A. Je to tabuľka, ktorá má toľko stĺpcov, koľko je tried podľa kódu $\mathcal{K}$ - p^{n-k} a toľko riadkov, koľko je kódových slov – p^k. V každom stĺpci tabuľky sú všetky slová jednej triedy, v prvom riadku tabuľky je reprezentant triedy. Po určení, v ktorom stĺpci sa dekódované slovo $\mathbf{w}$ nachádza dekódujeme tak, že od neho odčítame slovo v prvom riadku príslušného stĺpca.

	Trieda $\mathbf{e}_1 + \mathcal{K}$	Trieda $\mathbf{e}_2 + \mathcal{K}$		Trieda $\mathbf{e}_m + \mathcal{K}$
reprezentant	$\mathbf{e_1} = \mathbf{e_1} + \mathbf{o}$	$\mathbf{e_2} = \mathbf{e_2} + \mathbf{o}$	...	$\mathbf{e_m} = \mathbf{e_m} + \mathbf{o}$
prvky tried	$\mathbf{e_1} + \mathbf{u_1}$	$\mathbf{e_2} + \mathbf{u_1}$	...	$\mathbf{e_m} + \mathbf{u_1}$
	$\mathbf{e_1} + \mathbf{u_2}$	$\mathbf{e_2} + \mathbf{u_2}$	...	$\mathbf{e_m} + \mathbf{u_2}$
	...	...	...	...
	...	...	...	...
	...	...	...	...
	$\mathbf{e_1} + \mathbf{u_s}$	$\mathbf{e_2} + \mathbf{u_s}$	...	$\mathbf{e_m} + \mathbf{u_s}$

(1.110)

Slepianova tabuľka, $m = p^{n-k},\ s = |\mathcal{K}| = p^k$.

Slepianova tabuľka má p^n prvkov, ktoré v najhoršom prípade musíme prehľadať všetky. Pre bežne používané binárne kódy dĺžky 64 by to znamenalo v najhoršom prípade $2^{64} > 10^{19}$ prehľadaní. Šikovnou implementáciou možno úplné prehľadávanie nahradiť binárnym prehľadávaním, ktoré by v tomto prípade potrebovalo len 64 prístupov do tabuľky, ale nároky na príslušné údajové štruktúry ostávajú enormné.

Problém možno značne zredukovať, ak si uvedomíme, že všetky prvky triedy $\mathbf{e}+\mathcal{K}$ majú rovnaký syndrom ako jej reprezentant $\mathbf{e}$. Je to preto, lebo pre $\mathbf{v} \in \mathcal{K}$ a kontrolnú maticu $\mathbf{H}$ kódu $\mathcal{K}$ platí:

$$\mathbf{H}.(\mathbf{e} + \mathbf{v}) = \mathbf{H}.\mathbf{e} + \mathbf{H}.\mathbf{v} = \mathbf{H}.\mathbf{e} + \mathbf{o} = \mathbf{H}.\mathbf{e}.$$

Preto namiesto Slepianovej tabuľky stačí tabuľka s dvoma riadkami, kde v prvom riadku sú reprezentanti tried $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_m}$, $m = p^{n-k}$ a v druhom riadku sú príslušné syndromy $\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_m}$.

reprezentant	$\mathbf{e}_1$	$\mathbf{e}_2$	...	$\mathbf{e}_m$
syndrom	$\mathbf{s}_1$	$\mathbf{s}_2$	...	$\mathbf{s}_m$

(1.111)

Teraz možno štandardný dekódovací algoritmus preformulovať nasledovne: Pre prijaté slovo $\mathbf{w}$ vypočítame jeho syndrom $\mathbf{s} = \mathbf{H}.\mathbf{w}$. V tabuľke (1.111) nájdeme reprezentanta $\mathbf{e}$ triedy s rovnakým syndromom $\mathbf{s}$ a dekódujeme $\delta(\mathbf{w}) = \mathbf{w} - \mathbf{e}$.

Tabuľka (1.111) má p^{n-k} stĺpcov a len dva riadky – jej rozsah je podstatne menší ako rozsah Slepianovej tabuľky. Naviac možno očakávať, že ani pri veľkej dĺžke n kódu $\mathcal{K}$ sa nebude číslo $n - k$ príliš zvyšovať, pretože znamená tiež počet kontrolných znakov kódu, a ten sa z hľadiska udržania dobrého informačného pomeru snažíme zbytočne nezvyšovať.

1.14 Hammingové kódy

Veta 1.29. *p-znakový lineárny kód opravuje jednoduché chyby práve vtedy, keď žiaden stĺpec jeho kontrolnej matice nie je skalárnym násobkom iného stĺpca.*

Špeciálne binárny lineárny kód opravuje jednoduché chyby práve vtedy, keď stĺpce jeho kontrolnej matice sú nenulové a navzájom rôzne.

Dôkaz. Vieme, že kód $\mathcal{K}$ opravuje jednoduché chyby práve vtedy, keď $\Delta(\mathcal{K}) \geq 3$, čo podľa vety 1.23 (str.33) nastáva práve vtedy, keď ľubovoľné dva stĺpce kontrolnej matice $\mathbf{H}$ sú lineárne nezávislé.

Vo všeobecnom prípade sú dva vektory $\mathbf{u}$, $\mathbf{v}$ lineárne nezávislé práve vtedy, keď jeden nie je skalárnym násobkom druhého, čo v prípade binárnej abecedy je práve vtedy, keď sú oba vektory $\mathbf{u}$, $\mathbf{v}$ nenulové a rôzne. □

Definícia 1.27. Binárny lineárny (n, k)-kód sa nazýva **Hammingov kód**, ak jeho kontrolná matica $\mathbf{H}$ má za stĺpce všetky nenulové binárne slová dĺžky $n - k$, pričom každé z nich sa ako stĺpec matice $\mathbf{H}$ vyskytuje práve raz.

Ak sa majú v matici $\mathbf{H}$ všetky nenulové binárne slová dĺžky $n-k$ vyskytovať práve raz, musí byť počet stĺpcov v tejto matici rovný $n = 2^{(n-k)} - 1$. Preto môžu existovať len Hammingov (n, k)-kód pre $(n, k) = (3, 2),\ (7, 4),\ (15, 11),\ (31, 26), \ldots (2^m - 1, 2^m - m - 1), \ldots$. Všimnime si ešte, že informačný pomer (1.70) (str.22) s rastúcim m rýchlo rastie k 1. Napr. pre $m = 6$ Hammingov $(63, 57)$-kód má informačný pomer $\dfrac{57}{63} > 0.9$.

Definícia 1.28. Dekódovanie Hammingovho kódu. Predpokladajme, že stĺpce kontrolnej matice $\mathbf{H}$ sú usporiadané tak, že tvoria binárne rozvoje čísel $1, 2, \ldots, 2^{m-1}$. Prijmeme vektor $\mathbf{w}$ a vzpočítame jeho syndrom $\mathbf{s} = \mathbf{Hw}$. Ak $\mathbf{s} = \mathbf{o}$ slovo $\mathbf{w}$ nemeníme. Slovo $\mathbf{s}$ je binárnym rozvojom čísla i a my zmeníme i-tý znak prijatého slova $\mathbf{w}$, presnejšie

$$\delta(\mathbf{w}) = \begin{cases} \mathbf{w}, & \text{ak } \mathbf{s} = \mathbf{o} \\ \mathbf{w} - \mathbf{e_i}, & \text{ak } \mathbf{s} \text{ je binárnym rozvojom čísla } i, \end{cases} \tag{1.112}$$

kde $\mathbf{e}_i$ je slovo s jednotkou na mieste i.

Veta 1.30. *Dekódovanie δ definované v* (1.112) *opravuje jednoduché chyby. Presnejšie: Ak sa slovo* $\mathbf{w}$ *líši od niektorého kódového slova* $\mathbf{v}$ *nanajvýš v jednom znaku, potom* $\delta(\mathbf{w}) = \mathbf{v}$.

Dôkaz. Ak $\mathbf{w} = \mathbf{v}$, potom aj $\mathbf{w}$ je kódové slovo a platí $\mathbf{Hw} = \mathbf{Hv} = \mathbf{o}$, a v tom prípade $\delta(\mathbf{w}) = \mathbf{w} = \mathbf{v}$.

Nech sa slová $\mathbf{v}$, $\mathbf{w}$ líšia práve v jednom znaku, t.j. $\mathbf{w} = \mathbf{v} + \mathbf{e_i}$. Potom $\mathbf{Hw} = \mathbf{H}(\mathbf{v} + \mathbf{e_i}) = \mathbf{Hv} + \mathbf{He_i} = \mathbf{He_i}$. Ale $\mathbf{He_i}$ je i-tý stĺpec matice $\mathbf{H}$ a ten je rozvojom čísla i. Ak budeme dekódovať predpisom $\delta(\mathbf{w}) = \mathbf{w} - \mathbf{e_i} = \mathbf{v}$, budeme dekódovať správne. □

Medzi kódmi opravujúcimi t chýb sú najekonomickejšie tzv. perfektné kódy. Podľa definície 1.11 (str. 19) je blokový kód $\mathcal{K}$ dĺžky n t-perfektný, ak množina gulí $\{K_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ tvorí disjunkntý rozklad množiny A^n všetkých slov dĺžky n.

Veta 1.31. *Lineárny kód je t-perfektný práve vtedy, keď množina všetkých slov váhy menšej alebo rovnej než t tvorí systém všetkých reprezentantov všetkých tried slov podľa kódu $\mathcal{K}$.*

Dôkaz. Prv, než začneme dokazovať tvrdenie vety, si všimneme že ľubovoľné slovo $\mathbf{a} \in A^n$ môže byť reprezentantom niektorej triedy kódu $\mathcal{K}$ – totiž triedy $\mathbf{a} + \mathcal{K}$. Na to, aby sme dokázali, že množina všetkých slov váhy menšej alebo rovnej než t tvorí systém všetkých reprezentantov všetkých tried slov podľa kódu $\mathcal{K}$ stačí ukázať dve skutočnosti a to že

- každá trieda má reprezentanta s váhou menšou alebo rovnou t
- ak $\mathbf{e_1},\ \mathbf{e_2}$ sú dve slová také, že $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$, potom $\mathbf{e}_1 + \mathcal{K}$, $\mathbf{e}_1 + \mathcal{K}$ sú dve rôzne triedy, t.j. $\mathbf{e}_2 \notin (\mathbf{e}_1 + \mathcal{K})$

1. Nech je lineárny kód t-perfektný – t.j. pre každé slovo $\mathbf{a} \in A^n$ existuje práve jedno kódové slovo $\mathbf{b} \in \mathcal{K}$ také, že vzdialenosť slov $\mathbf{a},\ \mathbf{b}$ je menšia lebo rovná t, t.j. $d(\mathbf{a},\ \mathbf{b}) \leq t$. Označme $\mathbf{e} = \mathbf{a} - \mathbf{b}$. Pretože Hammingova vzdialenosť slov $\mathbf{a},\ \mathbf{b}$ je menšia lebo rovná t, je $\|\mathbf{e}\| \leq t$. Potom je $\mathbf{a} = \mathbf{e} + \mathbf{b}$. Každá trieda $\mathbf{a} + \mathcal{K}$ má reprezentanta $\mathbf{e}$ s váhou menšou alebo rovnou t.

Keby existovali dve slová $\mathbf{e}_1,\ \mathbf{e}_2$ také, že $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$ a $\mathbf{e}_2 \in (\mathbf{e}_1 + \mathcal{K})$, potom $\mathbf{e_2} - \mathbf{e_1} \in \mathcal{K}$ a $\|\mathbf{e_2} - \mathbf{e_1}\| \leq 2t$. Z poslednej nerovnosti vyplýva pre minimálnu vzdialenosť $\Delta(\mathcal{K})$ kódu $\mathcal{K}$ $\Delta(\mathcal{K}) \leq 2t$, čo je v spore s predpokladom, že $\mathcal{K}$ opravuje t chýb. Podľa vety 1.13 (str. 19) totiž kód $\mathcal{K}$ opravuje t chýb práve vtedy, keď $\Delta(\mathcal{K}) \geq 2t + 1$.

2. Nech množina všetkých slov váhy menšej alebo rovnej než t tvorí systém všetkých reprezentantov všetkých tried slov podľa kódu $\mathcal{K}$. Najprv ukážeme, že $\Delta(\mathcal{K}) \geq 2t + 1$. Keby totiž existovalo $\mathbf{a} \in \mathcal{K}$ také, že $\|\mathbf{a}\| \leq 2t + 1$, bolo by možné vyjadriť $\mathbf{a} = \mathbf{e_1} - \mathbf{e_2}$, kde $\|\mathbf{e}_1\| \leq t$, $\|\mathbf{e}_2\| \leq t$ a $\mathbf{e_1} \neq \mathbf{e_2}$. Podľa (i) vety 1.25 (str. 34) by potom $(\mathbf{e}_1 + \mathcal{K}) = (\mathbf{e}_2 + \mathcal{K})$ čo by bolo v spore s predpokladom, že $\mathbf{e_1},\ \mathbf{e_2}$ sú reprezentantmi rôznych tried. Ak je teda $\Delta(\mathcal{K}) \geq 2t + 1$, všetky gule $\{K_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ sú po dvoch disjunktné.

Teraz ukážeme, že pre každé $\mathbf{a} \in A^n$ existuje guľa $K_t(\mathbf{b})$, $\mathbf{b} \in \mathcal{K}$ taká, že $\mathbf{a} \in K_t(\mathbf{b})$. Podľa predpokladu existuje $\mathbf{e} \in A^n$, $\|\mathbf{e}\| \leq t$ také, že $\mathbf{a} \in (\mathbf{e} + \mathcal{K})$. Dá sa teda písať $\mathbf{a} = \mathbf{e} + \mathbf{b}$ pre nejaké $\mathbf{b} \in \mathcal{K}$. Odtiaľ $\mathbf{a} - \mathbf{b} = \mathbf{e}$ a preto $d(\mathbf{a}, \mathbf{b}) = \|(\mathbf{a} - \mathbf{b})\| = \|\mathbf{e}\| \leq t$ a teda $\mathbf{a} \in K_t(\mathbf{b})$. Systém gulí $\{K_t(\mathbf{a}) \mid \mathbf{a} \in \mathcal{K}\}$ tvorí disjunkntý rozklad množiny A^n, a preto je kód $\mathcal{K}$ t-perfektný. □

Veta 1.32. *Hammingové binárne kódy sú* 1*-perfektné. Každý* 1*-perfektný binárny lineárny kód je Hammingov.*

Dôkaz. Hammingov kód dĺžky $2^m - 1$ má m kontrolných znakov a podľa tvrdenia (iii) vety 1.25 (str. 34) má 2^m tried. Označme $\mathbf{e}_0 = \mathbf{o}$ – nulové slovo dĺžky $2^m - 1$. Ďalej označme pre $i = 1, 2, \ldots, 2^m - 1$

$$\mathbf{e}_i = \left[\ 0 \quad 0 \quad \ldots \quad 0 \quad 1 \quad 0 \ldots \quad 0\ \right],$$

t.j. vektor $\mathbf{e}_i$ má všade nuly okrem miesta i, na ktorom má znak 1. Všetky $\mathbf{e}_i$ pre $i = 1, 2, \ldots, 2^m - 1$ sú nekódové slová.

Skúmajme triedy $\mathbf{e}_i + \mathcal{K}$ pre $i = 0, 1, 2, \ldots, 2^m - 1$. Trieda $\mathbf{e}_0 + \mathcal{K}$ je totožná s množinou kódových slov $\mathcal{K}$ a je preto rôzna s ostatnými triedami. Keby boli dve triedy $\mathbf{e}_i + \mathcal{K}$, $\mathbf{e}_j + \mathcal{K}$ totožné pre $i \neq j$, potom by $\mathbf{e_i} - \mathbf{e_j} \in \mathcal{K}$, čo by znamenalo lineárnu závislosť i-teho a j-teho stĺpca kontrolnej matice kódu $\mathcal{K}$, (čo je v prípade binárneho kódu rovnosť príslušných stĺpcov). Hammingov kód má však kontrolnú maticu, v ktorej žiadne dva stĺpce nie sú rovnaké.

Pretože Hammingov kód $\mathcal{K}$ má 2^m tried a my sme ukázali, že všetky triedy typu $\mathbf{e}_i + \mathcal{K}$ pre $i = 0, 1, 2, \ldots, 2^m - 1$ sú rôzne (a je ich 2^m), nemôže existovať žiadna ďalšia trieda. Množina všetkých slov dĺžky ≤ 1 tvorí systém reprezentantov všetkých tried Hammingovho kódu $\mathcal{K}$ a preto je tento kód 1-perfektný.

Majme binárny lineárny kód $\mathcal{K}$ s m kontrolnými znakmi, ktorý je 1 perfektný. Podľa tvrdenia (iii) vety 1.25 (str. 34) má kód $\mathcal{K}$ 2^m tried. Nech má tento kód kontrolnú maticu $\mathbf{H}$ typu $n \times m$. Podľa vety 1.29 musia byť všetky stĺpce matice $\mathbf{H}$ nenulové a rôzne. Preto pre počet stĺpcov matice $\mathbf{H}$ platí $n \leq 2^m - 1$. Pretože je kód $\mathcal{K}$ perfektný, podľa vety 1.31 (str. 37) sú všetky binárne slová dĺžky n s váhou nula alebo jedna práve všetci reprezentanti tried. Takýchto slov je $n+1$ (nulové slovo a všetky slová typu $\mathbf{e}_i$ s práve jednou jednotkou na i-tom mieste). Je preto

$$n + 1 = 2^m$$

čiže

$$n = 2^m - 1.$$

Kontrolná matica kódu $\mathcal{K}$ je matica typu $(2^m - 1) \times m$ a jej stĺpce sú práve všetky rôzne binárne nenulové slová dĺžky m. $\mathcal{K}$ je teda Hammingovým kódom. □

Definícia 1.29. Rozšírený Hammingov binárny kód je binárny kód, ktorý vznikne rozšírením Hammingovho kódu o znak celkovej kontroly parity.

Rozšírený Hammingov kód je $(2^m, 2^m - m - 1)$-kód všetkých slov $\mathbf{v} = v_1 v_2 \ldots v_{2^m}$ takých, že $v_1 v_2 \ldots v_{2^m - 1}$ je kódové slovo Hammingovho kódu a $v_1 + v_2 + \cdots + v_{2^m} = 0$. Jeho minimálna váha je 4. Tento kód opravuje jednoduché chyby a objavuje trojnásobné chyby.

Poznámka 1.6. Veta 1.29 dáva návod, ako definovať p-znakový Hammingov kód. Je to kód s kontrolnou maticou $\mathbf{H}$ takou, že

(i) žiaden stĺpec nie je skalárnym násobkom iného stĺpca

(ii) každé nenulové slovo je skalárnym násobkom niektorého stĺpca matice $\mathbf{H}$.

Maticu $\mathbf{H}$ môžeme zostaviť napríklad zo všetkých stĺpcov rovnakej dĺžky takých, ktoré majú prvý nenulový znak rovný 1. Dá sa ukázať, že p-znakové Hammingové kódy majú mnohé vlastnosti rovnaké resp. analogické ako binárne Hammingové kódy. Tak napríklad všetky Hammingové kódy sú 1-perfektné.

1.15 Golayov kód

Označme $\mathbf{B}$ štvorcovú maticu typu 11×11, ktorej prvý riadok obsahuje binárne slovo 11011100010 a ostatné riadky vzniknú pravými rotáciami prvého riadku, t.j.

$$\mathbf{B} = \begin{bmatrix}
1 & 1 & & 1 & 1 & 1 & & & & 1 & \\
 & 1 & 1 & & 1 & 1 & 1 & & & & 1 \\
1 & & 1 & 1 & & 1 & 1 & 1 & & & \\
 & 1 & & 1 & 1 & & 1 & 1 & 1 & & \\
 & & 1 & & 1 & 1 & & 1 & 1 & 1 & \\
 & & & 1 & & 1 & 1 & & 1 & 1 & 1 \\
1 & & & & 1 & & 1 & 1 & & 1 & 1 \\
1 & 1 & & & & 1 & & 1 & 1 & & 1 \\
1 & 1 & 1 & & & & 1 & & 1 & 1 & \\
 & 1 & 1 & 1 & & & & 1 & & 1 & 1 \\
1 & & 1 & 1 & 1 & & & & 1 & & 1
\end{bmatrix} \tag{1.113}$$

Binárne slovo 11011100010 má na mieste i jednotku práve vtedy, keď je $i-1$ štvorcom modulo 11, t.j. ak $i-1 = 0^2$, 1^2, 2^2, 3^2, $4^2 \equiv 5$ a $5^2 \equiv 3$. V celej časti 1.15 budeme predpokladať, že matica $\mathbf{B}$ je daná vzťahom (1.113).

Definícia 1.30. Golayov kód G_{23} je systematický binárny kód dĺžky 23 s generujúcou maticou $\mathbf{G}_{23}$ definovanou

$$\mathbf{G}_{23} = \left[\begin{array}{c|c} \mathbf{E}_{12\times 12} & \begin{array}{c} \mathbf{B}_{11\times 11} \\ \hline 11\ldots11 \end{array} \end{array}\right], \tag{1.114}$$

kde $\mathbf{E}_{12\times 12}$ je jednotková matica typu 12×12, $\mathbf{B}_{11\times 11}$ je štvorcová matica typu 11×11 definovaná v (1.113).

Golayov kód G_{24} je systematický binárny kód dĺžky 24 s generujúcou maticou $\mathbf{G}_{24}$, ktorá vznikne z matice $\mathbf{G}_{23}$ pridaním stĺpca $11\ldots10$, t.j.

$$\mathbf{G}_{24} = \left[\begin{array}{c|c|c} \mathbf{E}_{12\times 12} & \begin{array}{c} \mathbf{B}_{11\times 11} \\ \hline 11\ldots11 \end{array} & \begin{array}{c} 1 \\ 1 \\ \ldots \\ 1 \\ \hline 0 \end{array} \end{array}\right] \tag{1.115}$$

$$\left[\begin{array}{cccccccccccccccccccccccc}
1 & & & & & & & & & & & 1 & 1 & & 1 & 1 & 1 & & & & 1 & & 1 \\
& 1 & & & & & & & & & & & 1 & 1 & & 1 & 1 & 1 & & & & 1 & 1 \\
& & 1 & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & & & 1 \\
& & & 1 & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & & 1 \\
& & & & 1 & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & & 1 \\
& & & & & 1 & & & & & & & & & 1 & & 1 & 1 & & 1 & 1 & 1 & 1 \\
& & & & & & 1 & & & & & 1 & & & & 1 & & 1 & 1 & & 1 & 1 & 1 \\
& & & & & & & 1 & & & & 1 & & & & & 1 & & 1 & 1 & & 1 & 1 \\
& & & & & & & & 1 & & & 1 & 1 & & & & & 1 & & 1 & 1 & & 1 \\
& & & & & & & & & 1 & & 1 & 1 & 1 & & & & & 1 & & 1 & 1 & 1 \\
& & & & & & & & & & 1 & & 1 & 1 & 1 & & & & & 1 & & 1 & 1 & 1 \\
& & & & & & & & & & & 1 & & 1 & 1 & 1 & & & & & 1 & & 1 & 1 \\
& & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{array}\right]$$

Generujúca matica Golayovho kódu $\mathbf{G}_{24}$.

Vlastnosti kódov G_{24}**,** G_{23}**.**

- Počet informačných znakov kódu G_{24} je 12, počet kontrolných znakov je tiež 12.
- Kód G_{24} je samoduálny – jeho kontrolná matica je aj jeho generujúcou maticou (na to stačí overiť, že skalárny súčin ľubovoľných dvoch riadkov matice $\mathbf{G}_{24}$ je rovný 0).
- Minimálna vzdialenosť kódu G_{24} je 8
- Kód G_{23} je $(23, 12)$-kód, ktorý je 3-perfektný.

Veta 1.33. Tietäväinen, Van Lint. *Jediné netriviálne perfektné binárne kódy sú tieto:*

a) Hammingove kódy pre jednoduché chyby,

b) Golayov kód G_{23} *pre trojnásobné chyby a kódy s ním ekvivalentné,*

c) opakovacie kódy dĺžky $2t + 1$ *pre* t*-násobné chyby, kde* $t = 1, 2, 3, \ldots$.

Dôkaz vlastností Golayových kódov i poslednej vety presahujú rámec tejto publikácie, preto ich neuvádzame. Na záver tejto kapitoly ešte uveďme, že existuje perfektný Golayov ternárny $(11, 6)$-kód opravujúci trojnásobné chyby. Jeho generujúca matica je v tvare

$$\mathbf{G}_{11} = \left[\begin{array}{c|c} \mathbf{E}_{6\times 6} & \mathbf{D}_{5\times 5} \\ \hline & 11\ldots11 \end{array}\right], \tag{1.116}$$

kde $\mathbf{E}_{6\times 6}$ je jednotková matica typu 6×6 a kde $\mathbf{D}_{5\times 5}$ je matica, ktorej riadky tvoria všetky cyklické pravé rotácie slova 01221. Okrem tohoto kódu (a kódov s ním ekvivalentných) sú jediné perfektné ternárne netriviálne kódy Hammingove a opakovacie kódy dĺžky $2t + 1$.

V prípade abecedy s viac ako troma znakmi sú jediné perfektné netriviálne kódy Hammingove a opakovacie kódy dĺžky $2t + 1$.

Na záver kapitoly o kódovaní treba povedať, že jej náplň tvorí iba úvod do teórie a praxe kódovania. Množstvo metód a kódov sa do tejto kapitoly nevošlo nielen z priestorových dôvodov, ale aj preto, lebo sú založené na pojmoch konečnej algebry ako boolovský polynom, okruhy plynómov, konečené telesá atď. Takými sú Reedove-Mullerove kódy, cyklické kódy, BCH kódy atď. Čitateľ však túto problematiku môže nájsť v použitej literatúre a znalosť tejto kapitoly mu značne pomôže orientovať sa v problematike kódovania.

a