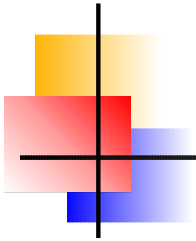
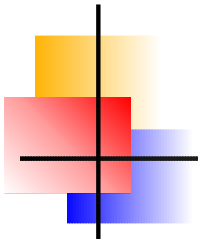


Kódovanie

Stanislav Palúch

Fakulta riadenia a informatiky, Žilinská univerzita

5. apríla 2020



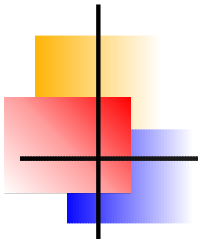
Prenosový reťazec

Všeobecná schéma prenosového reťazca je nasledujúca:

Dôvody kódovania

- Prispôsobenie abecedy zdroja abecede kanála
- Kompresia dát
- Ochrana proti chybám pri prenose
  - identifikácia istého počtu chýb
  - oprava istého počtu chýb

Ochrana údajov proti neautorizovaným zásahom nie je účelom kódovania



Abeceda, kód a kódovanie

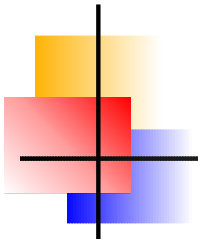
Nech $A = \{a_1, a_2, \ldots, a_r\}$ je konečná r-prvková množina. Prvky množiny A nazveme **znakmi**, množinu A **abecedou**. Množinu

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

nazveme **množinou všetkých slov abecedy** A.
Dĺžka slova a $\in A^*$ je počet znakov slova **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

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

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
$01010001 = 0101|0001 = 010|100|01 = 0|1|0|1|0|0|0|1.$



Abeceda, kód a kódovanie

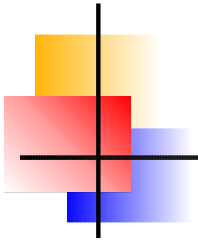
Nech $A = \{a_1, a_2, \ldots, a_r\}$ je konečná r-prvková množina. Prvky množiny A nazveme **znakmi**, množinu A **abecedou**. Množinu

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

nazveme **množinou všetkých slov abecedy** A.
Dĺžka slova a $\in A^*$ je počet znakov slova **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

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

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
$01010001 = 0101|0001 = 010|100|01 = 0|1|0|1|0|0|0|1.$



Abeceda, kód a kódovanie

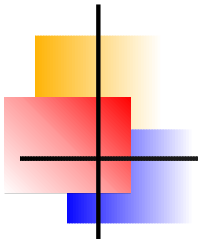
Nech $A = \{a_1, a_2, \ldots, a_r\}$ je konečná r-prvková množina. Prvky množiny A nazveme **znakmi**, množinu A **abecedou**. Množinu

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

nazveme **množinou všetkých slov abecedy** A.
Dĺžka slova a $\in A^*$ je počet znakov slova **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

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

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
$01010001 = 0101|0001 = 010|100|01 = 0|1|0|1|0|0|0|1.$



Abeceda, kód a kódovanie

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

$$K : A \to B^*,$$

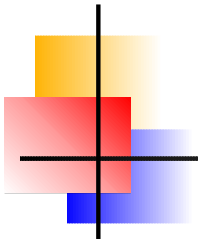
t. j. predpis, ktorý každému prvku abecedy A priradí slovo abecedy B.
Abeceda A je **zdrojová abeceda**, jej znaky sú **zdrojové znaky**, abeceda B je **kódová abeceda** a jej znaky sú **kódové znaky**.

Množinu všetkých slov kódovej abecedy typu

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

nazveme **kódom**, každé slovo z množiny $\mathcal{K}$ je **kódové slovo** ostatné slová z abecedy 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.



Abeceda, kód a kódovanie

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

$$K : A \to B^*,$$

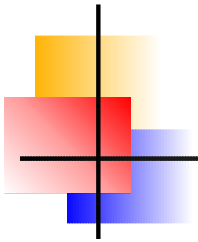
t. j. predpis, ktorý každému prvku abecedy A priradí slovo abecedy B.
Abeceda A je **zdrojová abeceda**, jej znaky sú **zdrojové znaky**, abeceda B je **kódová abeceda** a jej znaky sú **kódové znaky**.

Množinu všetkých slov kódovej abecedy typu

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

nazveme **kódom**, každé slovo z množiny $\mathcal{K}$ je **kódové slovo** ostatné slová z abecedy 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.



Abeceda, kód a kódovanie

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

$$K : A \to B^*,$$

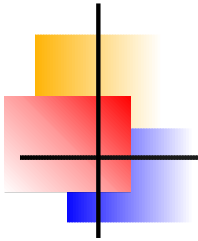
t. j. predpis, ktorý každému prvku abecedy A priradí slovo abecedy B.
Abeceda A je **zdrojová abeceda**, jej znaky sú **zdrojové znaky**, abeceda B je **kódová abeceda** a jej znaky sú **kódové znaky**.

Množinu všetkých slov kódovej abecedy typu

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

nazveme **kódom**, každé slovo z množiny $\mathcal{K}$ je **kódové slovo** ostatné slová z abecedy 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.



Abeceda, kód a kódovanie

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

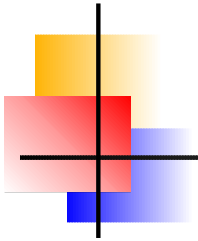
$$K^*(a_{i_1} a_{i_2} \ldots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_n})$$

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priradiť kódové slová rôznej dĺžky.

Často sa však stretávame s kódovaniami, 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.



Abeceda, kód a kódovanie

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

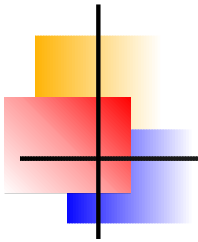
$$K^*(a_{i_1} a_{i_2} \ldots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_n})$$

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priradiť kódové slová rôznej dĺžky.

Často sa však stretávame s kódovaniami, 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.



Abeceda, kód a kódovanie

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

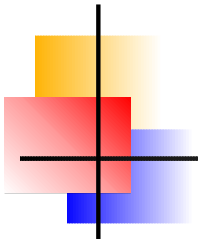
$$K^*(a_{i_1} a_{i_2} \ldots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_n})$$

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priradiť kódové slová rôznej dĺžky.

Často sa však stretávame s kódovaniami, 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.



Abeceda, kód a kódovanie

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

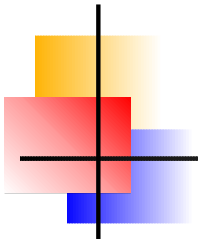
$$K^*(a_{i_1} a_{i_2} \ldots_{i_n}) = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_n})$$

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priradiť kódové slová rôznej dĺžky.

Často sa však stretávame s kódovaniami, 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.



Abeceda, kód a kódovanie

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

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

Kódovanie K^* je vlastne kódovaním znak po znaku.

Kódovanie môže rôznym znakom priradiť kódové slová rôznej dĺžky.

Často sa však stretávame s kódovaniami, 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

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.

Abeceda, kód a kódovanie

Príklad

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.*

Príklad

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

Hovoríme, že kódovanie $K : A \to B^*$ *je* **jednoznačne dekódovateľné**, *ak zo znalosti zakódovanej správy* $K^*(a_1 a_1 \dots a_n)$ *môžeme vždy určiť zdrojovú správu* $a_1 a_1 \dots a_n$, *t. j. ak je zobrazenie* $K^* : A^* \to B^*$ *prostým zobrazením.*

Príklad

Rozšírme zdrojovú abecedu z príkladu 2 na $A = \{1, 2, 3, 4, 5\}$ *a definujme kódovanie*

$$K(1) = 01,\ K(2) = 0,\ K(3) = 011,\ K(4) = 111,\ K(5) = 101.$$

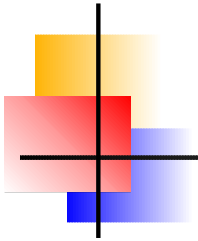
Majme správu 0101101. *Pre dekódovanie by sme ju mohli rozdeliť nasledovne:*

$$0|101|101,\quad 01|01|101,\quad 01|011|01,$$

pričom tieto delenia zodpovedajú zdrojovým slovám porade

$$255,\qquad 115,\qquad 131.$$

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.



Prefixové kódovanie a Kraftova nerovnosť

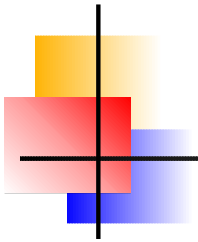
Definícia

Prefixom slova b $= b_1 b_2 \ldots b_k$ *nazveme každé zo slov*
$b_1, \quad b_1 b_2, \quad \ldots \quad , b_1 b_2 \ldots b_{k-1}, \quad b_1 b_2 \ldots b_k.$

Kódovanie resp. kód sa nazýva **prefixové**, *ak žiadne kódové slovo nie je prefixom iného kódového slova.*

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.



Prefixové kódovanie a Kraftova nerovnosť

Definícia

Prefixom slova b $= b_1 b_2 \ldots b_k$ *nazveme každé zo slov*
$b_1, \quad b_1 b_2, \quad \ldots \quad , b_1 b_2 \ldots b_{k-1}, \quad b_1 b_2 \ldots b_k.$

Kódovanie resp. kód sa nazýva **prefixové**, *ak žiadne kódové slovo nie je prefixom iného kódového slova.*

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.

Zobrazenie slov orientovným stromom

Majme abecedu A s n znakmi.

Vrcholy orienovaního stromu sú značené znakmi A. Každý vrchol je alebo list (vrchol bez následníkov), alebo má presne n následníkov označných znakmi abecedy A. Vrchol v reprezentuje slovo určené poradím snakov vrcholov cestu z koreňa do vrchola v.

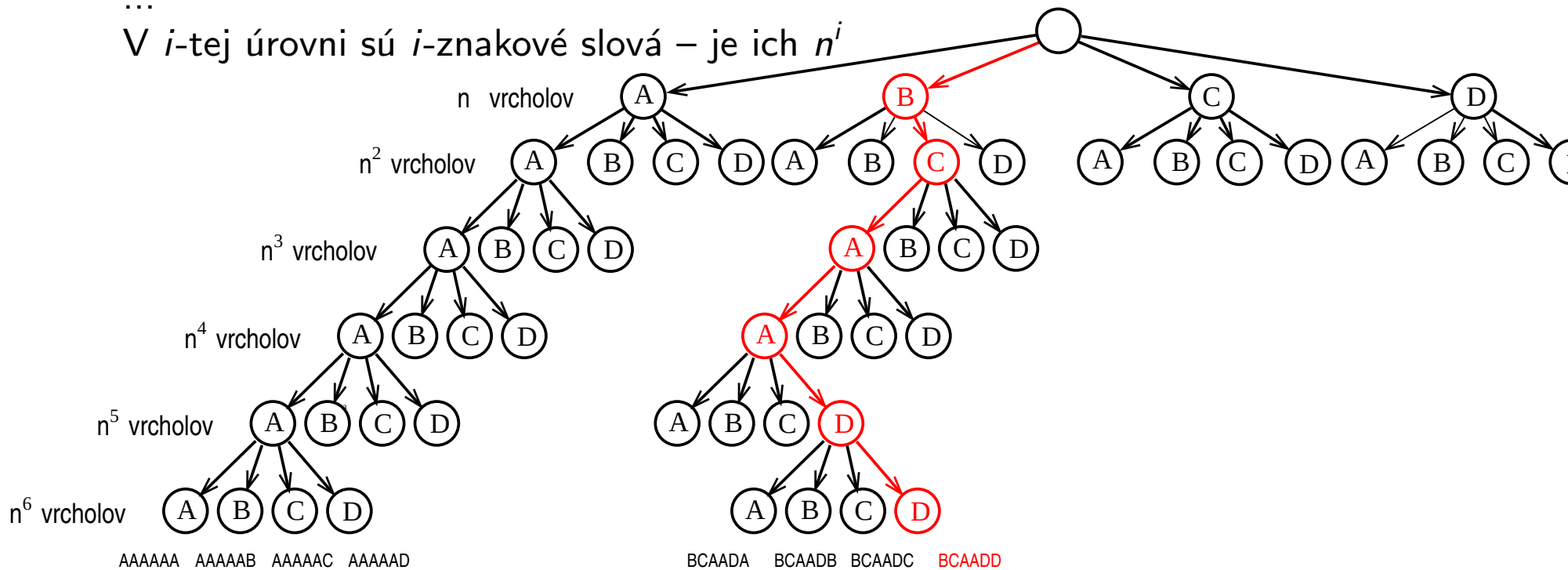
V nultej úrovni je koreň stromu

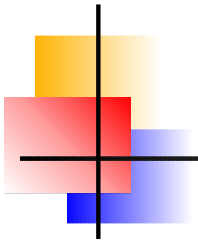
V prvej úrovni sú jednoznakové slová – je ich n

V druhej úrovni sú dvojznakové slová – je ich n^2

...

V i-tej úrovni sú i-znakové slová – je ich n^i





Kraftova nerovnosť

Veta

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

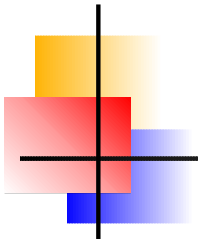
$$n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r} \leq 1. \tag{1}$$

Dôkaz.
Nech platí Kraftova nerovnosť (1).

Usporiadajme znaky zdrojovej abecedy tak, aby platilo

$$d_1 \leq d_2 \leq \cdots \leq d_r.$$

Za $K(a_1)$ zvoľme ľubovoľné slovo abecedy B dĺžky d_1.



Kraftova nerovnosť

Veta

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

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

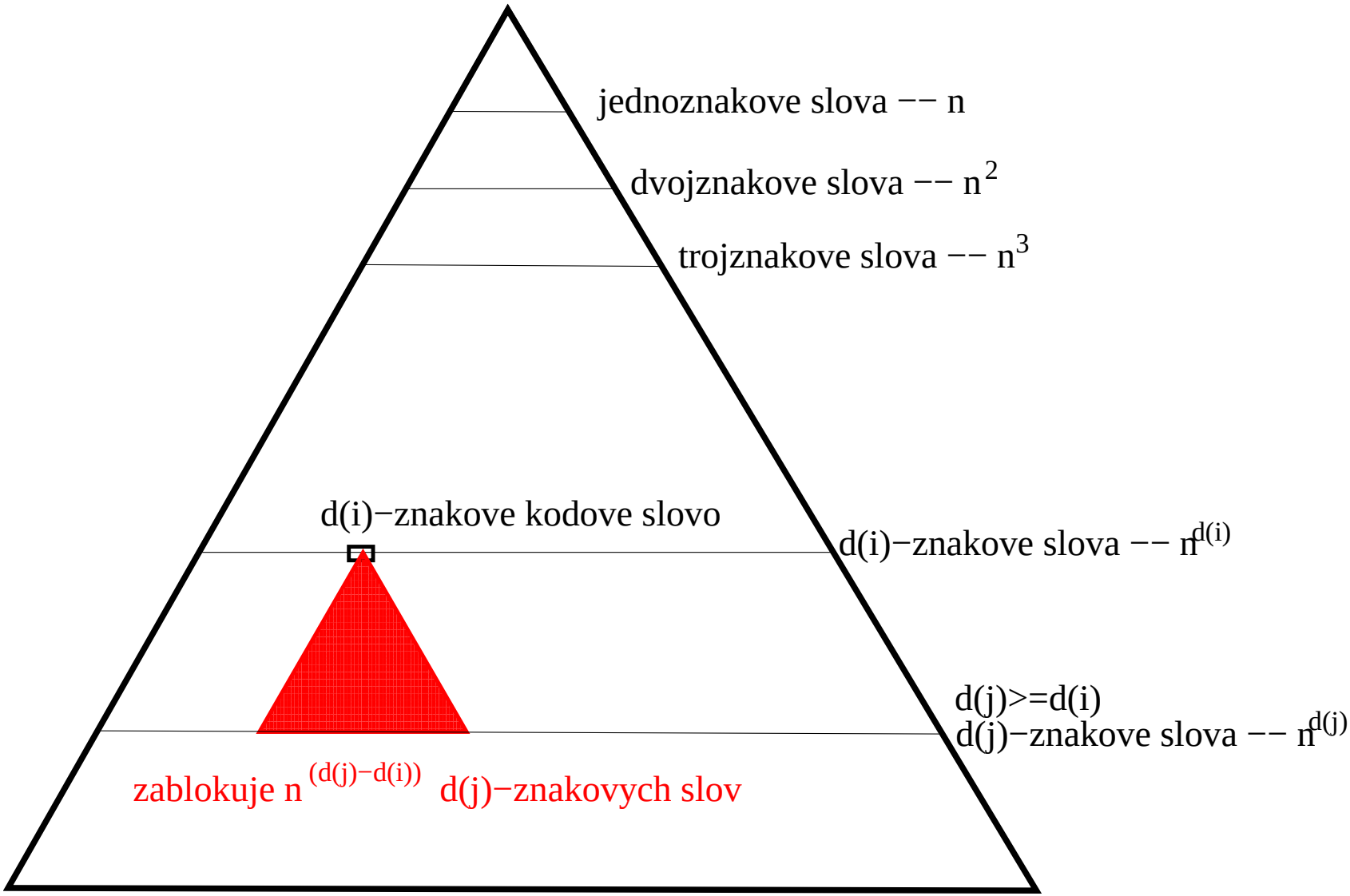
Dôkaz.
Nech platí Kraftova nerovnosť (1).

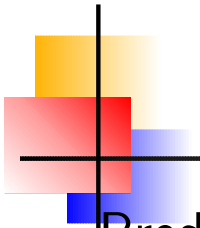
Usporiadajme znaky zdrojovej abecedy tak, aby platilo

$$d_1 \leq d_2 \leq \cdots \leq d_r.$$

Za $K(a_1)$ zvoľme ľubovoľné slovo abecedy B dĺžky d_1.

Kraftova nerovnosť





Kraftova nerovnosť

Predpokladajme, že už máme priradené kódové slová požadovanej dĺžky $K(a_1),\ K(a_2),\ \ldots,\ K(a_i)$.

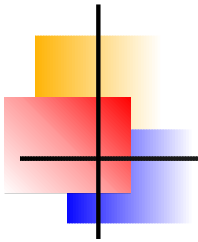
Pri voľbe kódového slova $K(a_{i+1})$ dĺžky d_{i+1} sa musíme vyhnúť

- $n^{(d_{i+1}-d_1)}$ 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

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \ldots + n^{(d_{i+1}-d_i)}. \qquad (2)$$



Kraftova nerovnosť

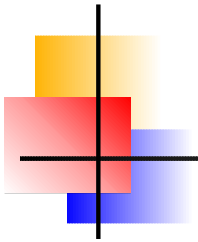
Keďže platí Kraftova nerovnosť (1), tým skôr platí pre prvých $i+1$ členov ľavej strany (1):

$$n^{-d_1} + n^{-d_2} + \cdots + n^{-d_i} + n^{-d_{i+1}} \leq 1. \tag{3}$$

Po vynásobení nerovnosti (3) číslom $n^{d_{i+1}}$ dostávame

$$n^{(d_{i+1}-d_1)} + n^{(d_{i+1}-d_2)} + \cdots + n^{(d_{i+1}-d_i)} + 1 \leq n^{d_{i+1}}. \tag{4}$$

Podľa (4) 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 $K(a_{i+1})$.



Kraftova nerovnosť

Majme prefixový kód s dĺžkami $d_1, d_2, \ldots, d_r$.

Predpokladajme

$$d_1 \leq d_2 \leq \cdots \leq d_r.$$

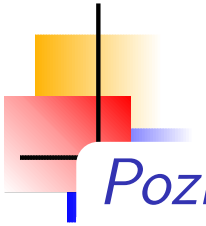
Existuje n^{d_r} slov dĺžky d_r, ktorými možno zakódovať písmeno a_r.

Pre každé $i = 1, 2, \ldots, r-1$ 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:

$$n^{(d_r - d_1)} + n^{(d_r - d_2)} + \cdots + n^{(d_r - d_{r-1})} + 1 \leq n^{d_r} \ . \qquad (5)$$

Vydelením nerovnosti (5) číslom n^{d_r} dostávame požadovanú Kraftovu nerovnosť (1). □



Kraftova nerovnosť

Poznámka

Algoritmus na zostrojenie prefixového kódu s dĺžkami slov

$$d_1, d_2, \ldots, d_r$$

Prvá časť dôkazu vety 1 je konštruktívna, dáva návod na zostrojenie prefixového kódovania, ak sú dané požadované dĺžky

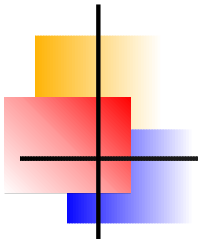
$$d_1 \leq d_2 \leq \cdots \leq d_r$$

kódových slov spĺňajúce 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 ako prefix žiadne zo slov $K(a_1), K(a_2), \ldots, K(a_i)$.

Existenciu aspoň jedného takéhoto slova zaručuje Kraftova nerovnosť.



Mac Millan-ova veta

Veta

Mac Millan.
Pre každé jednoznačne dekódovateľné kódovanie so zdrojovou abecedou

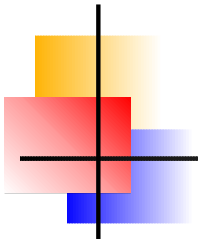
$$A = \{a_1, a_2, \ldots, a_r\}$$

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), *t.j.:*

$$n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r} \leq 1.$$



Mac Millan-ova veta

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

$$c = n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r} \ . \tag{6}$$

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.

Mac Millan-ova veta

Skúmajme výraz

$$c^k = \left[n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r}\right]^k = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_k=1}^{n} n^{-(d_{i_1}+d_{i_2}+\cdots+d_{i_k})} \ . \tag{7}$$

Pretože K je jednoznačne dekódovateľné, platí pre dve rôzne slová zdrojovej abecedy $a_1 a_2 \ldots a_{i_k}$, $a'_1 a'_2 \ldots a'_{i_k}$

$$K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k}) \neq K(a'_{i_1})|K(a'_{i_2})|\ldots|K(a'_{i_k}) \ .$$

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 (7) taký, že jeho záporne vzatý exponent $(d_{i_1} + d_{i_2} + \cdots + d_{i_k})$ sa rovná dĺžke slova $\mathbf{b}$.

Mac Millan-ova veta

Skúmajme výraz

$$c^k = \left[n^{-d_1} + n^{-d_2} + \cdots + n^{-d_r}\right]^k = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_k=1}^{n} n^{-(d_{i_1}+d_{i_2}+\cdots+d_{i_k})} \ . \tag{7}$$

Pretože K je jednoznačne dekódovateľné, platí pre dve rôzne slová zdrojovej abecedy $a_1 a_2 \ldots a_{i_k}$, $a'_1 a'_2 \ldots a'_{i_k}$

$$K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k}) \neq K(a'_{i_1})|K(a'_{i_2})|\ldots|K(a'_{i_k}) \ .$$

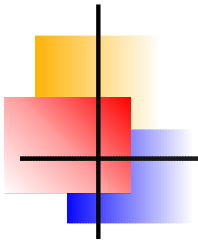
Preto ku každému slovu

$$\mathbf{b} = K(a_{i_1})|K(a_{i_2})|\ldots|K(a_{i_k})$$

z množiny $\mathcal{M}_k$ možno priradiť práve jeden sčítanec

$$n^{-(d_{i_1}+d_{i_2}+\cdots+d_{i_k})}$$

na pravej strane (7) taký, že jeho záporne vzatý exponent $(d_{i_1} + d_{i_2} + \cdots + d_{i_k})$ sa rovná dĺžke slova $\mathbf{b}$.



Mac Millan-ova veta

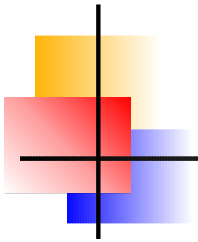
Ako sme už ukázali, maximálna dĺžka slova z množiny $\mathcal{M}_k$ je kd_r. Označme $M = kd_r$. Výraz na pravej strane vzťahu (7)

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

je polynómom stupňa M premennej $\frac{1}{n}$, a preto ho môžeme zapísať v tvare

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

V súčte na pravej strane posledného výrazu 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 $s_i \leq n^i$.



Mac Millan-ova veta

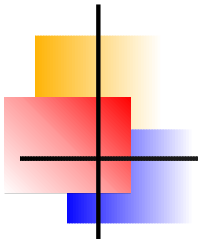
S využitím $s_i \leq n^i$ môžeme písať:

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

a teda

$$\frac{c^k}{k} \leq d_r \ . \tag{8}$$

Pretože nerovnosť (8) musí platiť pre ľubovoľné k, musí byť $c \leq 1$. □



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

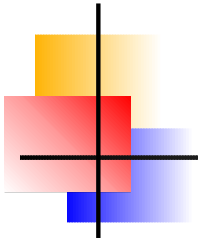
$$K(a_1),\ K(a_2),\ \ldots,\ K(a_r)$$

sú

$$d_1, d_2, \ldots, d_r.$$

Potom **stredná dĺžka kódového slova** kódovania K je

$$d(K) = p_1.d_1 + p_2.d_2 + \cdots + p_r.d_r = \sum_{i=1}^{r} p_i.d_i\ . \tag{9}$$



Najkratší kód - Huffmanova konštrukcia

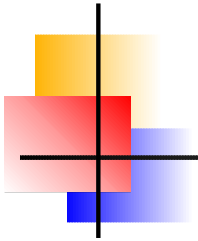
Ak kódovaním K kódujeme správu s veľkým počtom N znakov, môžeme očakávať, že dĺžka (počet znakov) zakódovanej správy v abecede B bude približne $N.d(K)$.

Keďže veľmi často (z hľadiska prenosu alebo uloženia správy) chceme, aby zakódovaná správa bola čo najkratšia, hľadáme kódovanie K s minimálnou strednou dĺžkou kódového slova $d(K)$.

Definícia

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.



Najkratší kód - Huffmanova konštrukcia

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 minimálnou strednou dĺžkou kódového slova $d(K)$.

Definícia

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.

Najkratší kód - Huffmanova konštrukcia

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 minimálnou strednou dĺžkou kódového slova $d(K)$.

Definícia

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.

Najkratší kód - Huffmanova konštrukcia

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 minimálnou strednou dĺžkou kódového slova $d(K)$.

Definícia

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.

Algoritmus na zostrojenie Huffmanovho binárneho kódu

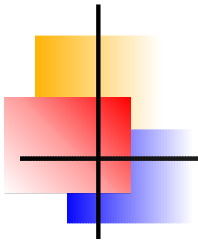
Budeme postupne budovať binárny koreňový strom, ktorého listy budú znaky zdrojovej abecedy A. Každý vrchol stromu bude mať priradenú pravdepodobnosť a binárny znak 0 alebo 1

Krok 1: Zostroj vrcholovo ohodnotený graf $G = (V, H, p)$, kde $V = A$ a kde $p(v)$ je pravdepodobnosť znaku v. Inicializačne polož $H := \emptyset$. Všetky vrcholy z V inicializačne prehlás za neoznačené.

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

Krok 3: Ak je graf G súvislý, choď na Krok 4, inak pokračuj Krokom 2.

Krok 4: Teraz je graf G koreňovým stromom s listami (t. j. vrcholmi stupňa 1) 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, postupnosť binárnych značiek vrcholov na tejto ceste určuje prefixový kód príslušného znaku.



Entropia zdroja a dĺžka najkratšieho kódovania

Entropia zdroja informácie

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

Pre stacionárny nezávislý zdroj $\mathcal{Z}$ s r-prvkovou abecedou $Z = \{a_1, a_2, \ldots, a_r\}$ s pravdepodobnosťami znakov $p_1, p_2, \ldots, p_r$

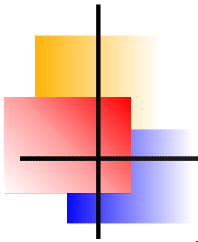
$$H(\mathcal{Z}) = -\sum_{i=1}^{r} p_1.\log_2(p_i) \ .$$

Majme ľubovoľné binárne prefixové kódovanie K abecedy Z 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 pre prípad stacionárneho nezávislého zdroja.

Entropia zdroja a dĺžka najkratšieho kódovania

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



Entropia zdroja a dĺžka najkratšieho kódovania

Zvoľme teraz prirodzené čísla d_i pre $i = 1, 2, \ldots, r$ tak, aby platilo

$$\log_2 \left(\frac{1}{p_i}\right) \leq d_i < log_2 \left(\frac{1}{p_i}\right) + 1$$

pre každé i.
Potom prvú nerovnosť môžeme postupne prepísať

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

Keďže posledná nerovnosť v predchádzajúcom riadku platí pre každé $i \in \{1, 2, \ldots, r\}$, 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 K s dĺžkami kódových slov $d_1, d_2, \ldots, d_r$.

Entropia zdroja a dĺžka najkratšieho kódovania

Prirodzené čísla d_i pre $i = 1, 2, \ldots, r$ boli zvolené tak, aby platilo

$$\log_2\left(\frac{1}{p_i}\right) \leq d_i < log_2\left(\frac{1}{p_i}\right) + 1 = -log_2\left(p_i\right) + 1$$

pre každé i.
Pre strednú dĺžku slova kódovania K platí:

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

Veta

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:

$$\boxed{H(\mathcal{Z}) \leq d_{\mathrm{opt}} < H(\mathcal{Z}) + 1\ .} \qquad (10)$$

Entropia zdroja a dĺžka najkratšieho kódovania

Majme stacionárny nezávislý zdroj $\mathcal{Z}$ so zdrojovou abecedou $Z = \{x, y, z\}$ s troma znakmi, ktorých pravdepodobnosti výskytu sú

$$p_x = 0.9, \quad p_y = 0.05, \quad p_z = 0.05$$

Najkratšie binárne prefixové kódovanie abecedy Z

$$K(x) = 0, \quad K(y) = 10, \quad K(z) = 11$$

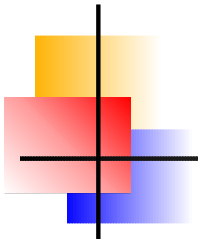
$$d(K) = 1 \times 0.9 + 2 \times 0.05 + 2 \times 0.05 = 1.1.$$

Entropia zdroja $\mathcal{Z}$ je $H(\mathcal{Z}) = 0.394$ bitu na znak.
Ak máme dostatočne dlhý N-znakový zdrojový text, potom dĺžku príslušného zakódovaného textu možno odhadnúť číslom

$$N \times 1.1,$$

jej dolná hranica hranica určená podľa vety 3 je $N \times 0.394$.

Dlhý zakódovaný zdrojový text bude teda v tomto prípade o 179% – t.j. skoro 3-krát dlhší ako dolný odhad jeho dĺžky určený entropiou $H(\mathcal{Z})$.



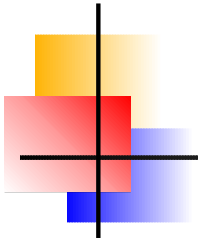
Entropia zdroja a dĺžka najkratšieho kódovania

Kódovanie znak po znaku však nie jediným spôsobom, ako zakódovať zdrojový text.

K zdroju $\mathcal{Z}$ s entropiou $H(\mathcal{Z})$ sme definovali 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 $\mathcal{Z}_{(k)}$ tiež stacionárnym nezávislým zdrojom.

Pre strednú dĺžku $d_{\text{opt}}^{(k)}$ kódového slova najkratšieho binárneho prefixového kódovania abecedy Z^k platí vzťah (10) z vety 3:

$$\begin{aligned} H(\mathcal{Z}_{(k)}) \leq & d_{\text{opt}}^{(k)} < H(\mathcal{Z}_{(k)}) + 1 \\ k.H(\mathcal{Z}) \leq & d_{\text{opt}}^{(k)} < k.H(\mathcal{Z}) + 1 \\ H(\mathcal{Z}) \leq & \frac{d_{\text{opt}}^{(k)}}{k} < H(\mathcal{Z}) + \frac{1}{k} \end{aligned} \tag{11}$$



Entropia zdroja a dĺžka najkratšieho kódovania

Veta

Základná veta o kódovaní zdrojov. *Nech $\mathcal{Z} = (Z^*, P)$ je stacionárny nezávislý zdroj s entropiou $H(\mathcal{Z})$.*
Potom je stredná dĺžka zakódovaného binárneho textu pripadajúca na jeden znak zdrojovej abecedy Z zdola ohraničená entropiou $H(\mathcal{Z})$.

Pritom sa dá nájsť prirodzené číslo k a binárne prefixové kódovanie slov zo $Z_{(k)}$ také, že stredná dĺžka zakódovaného textu pripadajúca na jeden znak zdrojovej abecedy Z je ľubovoľne blízko entropii $H(\mathcal{Z})$.