

Komprese

Cíl komprese: redukovat objem dat za účelem

- Přenosu dat
- Archivace dat
- Vytvoření distribuce sw
- Ochrana před viry

Kvalita komprese

- Rychlost komprese
- symetrie/asymetrie kompresního algoritmu
- kompresní poměr = poměr objemu komprimovaných dat k objemu nekomprimovaných dat

Komprese:

- bezztrátová – po kódování a dekódování je výsledek 100%shodný
- ztrátová po kódování a dekódování dochází ke ztrátě

Metody:

- jednoduché – založené na kódování opakujících se posloupností znaků (RLE)
- statistické – založené na četnosti výskytu znaků v komprimovaném souboru (Huffmanovo, Aritmetické)
- slovníkové – založené na kódování všech vyskytujících se posloupností (LZW)
- transformační – založené na ortogonálních popř. jiných transformacích (JPEG, waveletová, fraktálová)

RLE (Run Length Encoding) – kódování délkou běhu

Př:

- vstup: AAAABBCDDDDABD
- výstup: 4A2B1C4D1A1B1D

nevýhoda: pokud se znaky příliš neopakují nebude to komprese, ale naopak.

LZW (Lempel-Ziv-Welch) metoda

Princip:

- vyhledávání opakujících se posloupností znaků, ukládání těchto posloupností do slovníku pro další použití a přiřazení jednoznakového kódu těmto posloupnostem.
- jednoprůchodová metoda (nevyžaduje předběžnou analýzu souboru)
- Kódované znaky musí mít délku (počet bitů) větší než délka původních znaků (např. pro ASCII znaky (8 bitů) se obvykle

používá nová délka znaků 12 bitů popř. větší.)

- Při průchodu komprimovaným souborem se vytváří slovník (počet položek slovníku odpovídá hodnotě $2^{(\text{počet bitů nového kódu})}$, kde prvních $2^{(\text{počet bitů původního kódu})}$ položek jsou znaky původní abecedy a zbývající položky tvoří posloupnosti znaků obsažené v komprimovaném souboru.

Algoritmus komprese a vytvoření slovníku

```

s := přečti znak ze vstupu;
while (jsou další znaky na vstupu) do
begin
  C := přečti znak ze vstupu;
  if S+C je v kódovací tabulce then
    S := S+C
  else begin
    zapiš na výstup kód pro S
    přidej do kódovací tabulky (S+C)
    S := C
  end;
end;
zapiš na výstup kód pro S;

```

Příklad: Komprese řetězce ABCABCBCDABC

Postup kódování

S (prefix)	C (suffix)	výstup (kód)
A	B	A(65)
B	C	B(66)
C	A	C(67)
A	B	—
AB	C	AB(256)
C	A	—
CA	B	CA(258)
B	C	—
BC	D	BC(257)
D	A	D(68)
A	B	—
AB	C	—
ABC	—	ABC(259)

Výsledný výstupní řetězec:

65 66 67 256 258 257 68 259

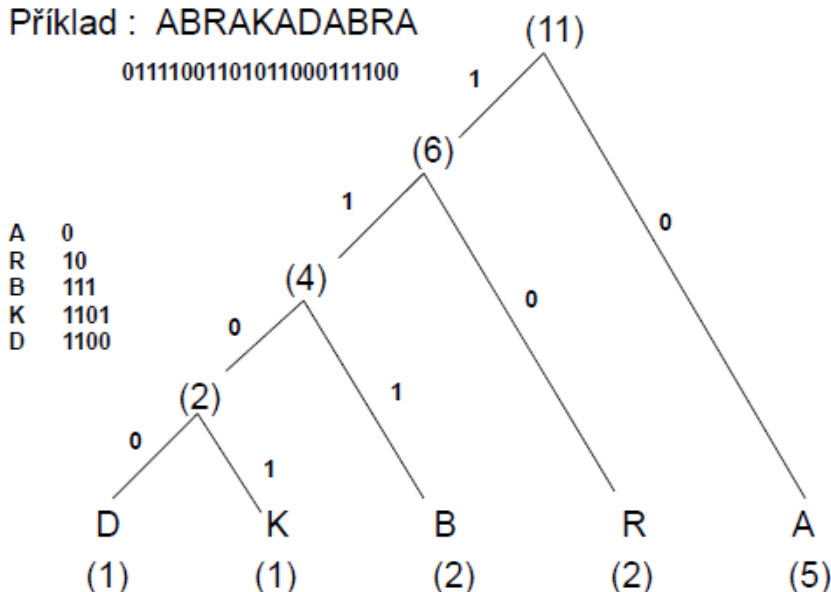
kód	posloupnost
0..255	jednotlivé znaky
256	AB
257	BC
258	CA
259	ABC
260	CAB
261	BCD
262	DA

Huffmanovo kódování

- využívá optimálního (nejkratšího) prefixového kódu

Příklad : ABRAKADABRA

01111001101011000111100



Aritmetické kódování

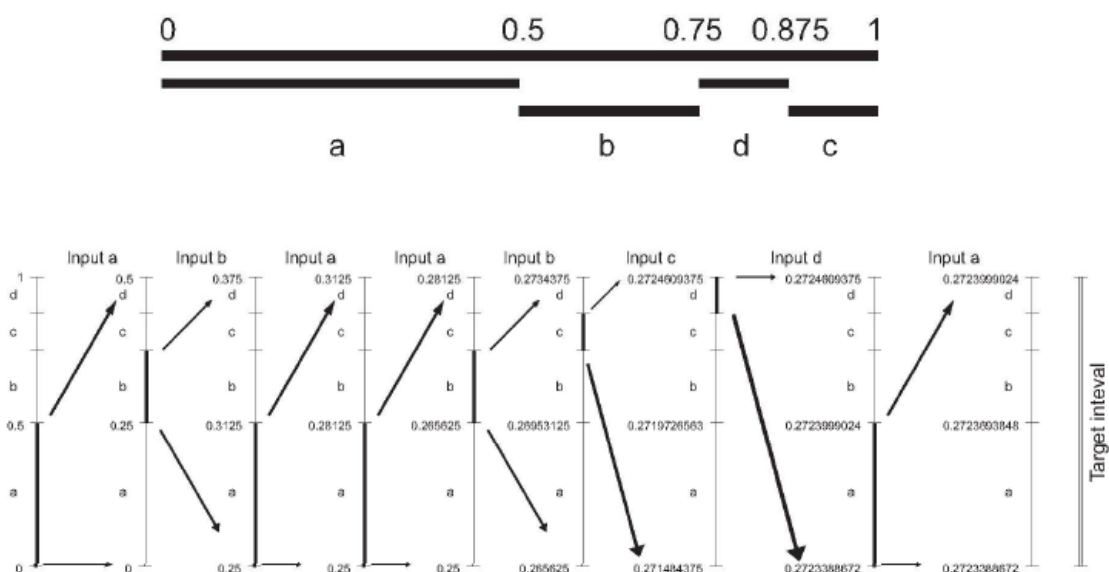
Princip : Aritmetické kódování reprezentuje zprávu jako podinterval intervalu $<0,1)$. Na začátku uvažujeme celý tento interval. Jak se zpráva prodlužuje, zpřesňuje se i výsledný interval a jeho horní a dolní mez se k sobě přibližují. Čím je kódovaný znak pravděpodobnější, tím se interval zúží méně a k zápisu delšího (to znamená hrubšího) intervalu stačí méně bitů. Na konec stačí zapsat libovolné číslo z výsledného intervalu.

-Kóduje celou zprávu jako jedno kódové slovo (v původní verzi číslo z intervalu $[0,1)$).

1. Zjištění pravděpodobnosti $P(i)$ výskytu jednotlivých znaků ve vstupním souboru
2. Stanovení příslušných kumulativních pravděpodobností $K(0)=0, K(i)=K(i-1)+P(i-1)$ a rozdělení intervalu $<0,1)$ na podintervaly $I(i)$ odpovídající jednotlivým znakům (seřazeným podle abecedy) tak, aby délky těchto intervalů vyjadřovaly pravděpodobnosti příslušných znaků: $I(i) = <K(i), K(i+1))$
3. Uložení použitých pravděpodobností
4. Vlastní komprese

Příklad kódování

$P(a)=0.5, P(b)=0.25, P(c)=0.125, P(d)=0.125$



JPEG (Join Photographic Experts Group)

- patří mezi nejvíce používané komprese u obrázků
- je vhodná pro komprimaci fotek, nevhodná pro např. technické výkresy (čárové výkresy) – dochází k viditelnému rozmazání

Princip:

- části obrazu se transformují do frekvenční oblasti (výsledkem je matice „frekvenčních“ koeficientů)
- z matice koeficientů se odstraní koeficienty odpovídající vyšším frekvencím
- zbývající koeficienty se vhodným způsobem zkomprimují