Projekt 1a – Kódování a komprese dat (KKO), letní semestr 200/2008

ahed

Jaroslav Bartoň, xbarto42@stud.fit.vutbr.cz

Funkce:

Program umí zakódovat a dekódovat vstup za pomoci adaptivního Huffmanova kódování. Kódování probíhá v jednom průchodu, lze ji tedy použít i na standardní vstup.

Parametry příkazové řádky:

Parametry příkazové řádky jsou zpracovávány pomocí knihovny *getopts*.

-c	Zakóduj vstup
-x	Dekóduj vstup
-i vstupní_soubor	Určení vstupního souboru, pokud není zadán, použije se standardní vstup
-o výstupní_soubor	Určení výstupního souboru, pokud není zadán, použije se standardní výstup
-l logovací_soubor	Určení souboru pro uložení výsledků kódování. Výpis je ignorován, pokud není zadán.
-h	Zobrazení nápovědy

Testování a výsledky:

Soubor	Původní velikost	Kódovaná velikost	Doba kódování	Doba dekódování	Komprese
zdrojové kódy kodéru	20 514 B	13 870B	212ms	210ms	33%
zvukový soubor mp3	33 387 731 B	33 268 083 B	8m 8,678s	8m 5,534s	0,3%
binární soubor ahed	55 152 B	40 104 B	696ms	692ms	27,3%
binární soubor /bin/tar	327 712 B	249 314 B	3,824s	3,740s	23,9%
textový soubor lorem…	32 392 B	17 442 B	280ms	280ms	46,2%

Popis algoritmu:

Program využívá adaptivního Huffmanova kódování založeném na algoritmu FKO (Faller-Gallager-Knuth). Huffmanův kód je kód s proměnnou délkou kódového slova tak aby znaky s vyšší četností výskytu měly kratší kód. Proto musí být prefixový tak aby prefix kódového slova neodpovídal žádnému jinému kódovému slovu. Kód každého znaku je určen z pozice znaku ve stromě, kde počínaje kořenem má levý potomek má hodnotu bitu kódu 0 a pravý potomek 1. Strom je sestavován v průběžně jak jsou načítány vstupní znaky.

Pro každý vstupní znak je počítán počet výskytů. Nutností znovu sestavit strom při změně četnosti je zabráněno použitím algoritmu FGK. Tento algoritmus začíná

pouze s uzlem označeným jako ZERO. Do tohoto uzlu se budou připojovat nově vytvořené podstromy, přičemž jako levý potomek se připojí ZERO, pravý potomek je pak nově načtený znak. Každý uzel má svoje hodnocení, které je rovnu součtu hodnocení potomků. Je třeba také kontrolovat, zda nedošlo k porušení sourozenecké vlastnosti.

Binární strom má sourozeneckou vlastnost jestliže každý uzel (s vyjímkou kořenu) má sourozence a pokud lze uzly seřadit do monotónní posloupnosti (podle četnosti daného uzlu) tak, že má každý uzel v posloupnosti za souseda svého sourozence. Pokud je zjištěno, že byla sourozenecká vlastnost porušena, je třeba přeuspořádat strom tak, aby byla znovu obnovena.

Pseudokód[1]:

Zpracování vstupního souboru:

```
načtiZnak X
dokud nejsi na konci souboru:
  pokud jsi X načetl poprvé:
    zapiš kód uzlu ZERO
    zapiš X
    vytvoř nový uzel U s následníky ZERO a listem X
    aktualizuj_strom od uzlu U
  znak X jsi již viděl:
    zapiš kód uzlu X
    aktualizuj strom od uzlu X
  načti znak X
```

Aktualizace stromu:

```
dokud U není kořen stromu:
  pokud existuje uzel U1 se stejným ohodnocením a vyšším pořadím:
    vyměň U1 a U
  zvyš ohodnocení U o 1
  U nastav předka U
zvyš ohodnocení U o 1
```

Použité zdroje:

[1] Paulíček, M.: Adaptivní Huffmanovo kódování – FGK. [online; citováno 16. 4. 2008], url: http://www.stringology.org/DataCompression/fgk/index_cs.html

Dodatek na závěr:

Vzhledem k tomu, že zadání je jasné, přesně specifikované a neměnné, doufám, že mi ho podařilo splnit. Kdyby takto byly zadány všechny projekty, neměl bych si na co stěžovat. Bohužel jedno zadání se letos změnilo a upřesňovalo několikrát.

1 vnořené bloky kódu jsou značeny odsazováním (například jazyk Python)