

KKO - Dokumentace

Projekt č. 1

Pelikán Jakub, `xpelik14@stud.fit.vutbr.cz`

1 Aplikace - `main.c`

Aplikace podle zadaných parametrů kóduje (parametr `-c`) nebo dekóduje (parametr `-x`) pomocí adaptivního Huffmanova kódování vstupní soubor. Vstupní soubor (parametr `-i`) a výstupní soubor (parametr `-o`) jsou zadány pomocí příslušného parametru následovaného názvem souboru. Pokud není zadán parametr `-i`, za vstup je považován `stdin`, pokud není zadán parametr `-o`, výstup je vypsán na `stdout`. Výstupní zpráva se zadává pomocí parametru `-l` následovaného názvem souboru, do kterého má být výstupní zpráva zapsána. Pokud není parametr `-l` zadán, tak je výpis výstupní zprávy ignorován. Ke zpracování parametrů je použita funkce `getopt_long`.

Vzájemná kombinace parametrů `-c -x` nebo duplikace libovolného parametru je považována za chybné zadání parametrů a vede k ukončení programu.

2 Knihovna - `ahed.c`

Knihovna obsahuje dvě hlavní funkce `AHEDEncoding` pro kódování a `AHEDDecoding` pro dekódování vstupního souboru a pomocné funkce, které jsou využívány hlavními funkcemi.

Při implementaci jsou použity 3 typy uzlů, které se vyskytují v Huffmanově stromě.

- **Escape** - uzel jehož kód představuje speciální escapovací symbol, který se vkládá do kódovaného výstupu před první výskyt symbolu.
- **Char** - uzel jehož kód reprezentuje určitý symbol.
- **NoChar** - vnitřní uzel stromu, který nerepresentuje určitý symbol.

2.1 Kódování

Na začátku procesu kódování se vytvoří **escape** uzel, který je zároveň kořenem stromu. Poté se čte vstupní soubor po znacích, kdy jako znak se bere 8 bitový ASCII znak. Po přečtení znaku se vyhledá ve vytvořeném Huffmanově stromě, zda obsahuje **char** uzel s načteným symbolem, pokud ano, tak proběhne inkrementování **char** uzlu 2.3. Pokud symbol nebyl ve Huff. stromě nalezen, tak se do výstupního stromu zapíše kód **escape** uzlu 2.6, za kterým následuje čtený znak. Poté se do Huff. stromu přidá nový uzel pro daný symbol 2.5. Po přečtení celého vstupu se na výstup zapíše kód **escape** uzlu 2.6 a doplní se posloupností jedničkových bitů tak, aby se do souboru zapsal celý bajt.

2.2 Dekódování

Na začátku procesu kódování se vytvoří **escape** uzel, který je zároveň kořenem stromu. Při procesu dekódování se postupuje podle typu aktuálního uzlu.

- Uzel **char** zapíše znak, který reprezentuje na výstup, inkrementuje daný uzel 2.3 a algoritmus pokračuje v kořenovém uzlu stromu.
- Uzel **escape** přečte 8 bitů ze vstupu, pokud již vstupní soubor neobsahuje dalších 8 bitů, ale méně a všechny bity mají jedničkovou hodnotu, pak se ukončí dekodování jako úspěšné. Pokud nejsou bity jedničkové, pak je to neočekávaný konec souboru a dekodování skončí s chybou. Pokud za uzlem je 8 bitů, tak se přidá **char** uzel 2.5, který reprezentuje přečtený 8-bit znak, a zároveň se zapíše znak do výstupního souboru. Algoritmus pokračuje v kořenu stromu.
- Uzel **noChar** přečte jeden bit ze vstupu a pokud je daný bit jedničkový, pak algoritmus pokračuje v uzlu, který je pravým synem aktuálního uzlu, jinak algoritmus pokračuje v uzlu, který je levým synem aktuálního uzlu.

2.3 Inkrementování uzlu

Nejprve se prochází uzly, které mají stejné ohodnocení jako inkrementovaný uzel, z těchto uzlů se vybere ten, který je umístěn na úrovni, která je nejbližší ke kořenu stromu, a v této úrovni je na nejpravější pozici. Pokud vybraný uzel není zároveň i inkrementovaným uzlem, pak se prohodí inkrementovaný uzel za vybraný uzel. Následně se inkrementuje hodnota čítače znaků inkrementovaného uzlu. Poté se zkontroluje, zda hodnota čítače uzlu nedosáhla maximální hodnoty, kterou lze do proměnné čítače uložit. Pokud ano, provede se změna měřítka stromu 2.4. Pokud inkrementovaný uzel není kořenem stromu, tak proces inkrementování opakujeme, ale inkrementovat se bude uzel, který je rodičem inkrementovaného uzlu.

2.4 Změna měřítka stromu

Při změně měřítka se vydělí čítače znaků listových (**char**) uzlů dvěma. Poté se aktualizují hodnoty čítačů vnitřních uzlů a to tak, že nová hodnota čítače je dána jako součet hodnot aktualizovaných čítačů levého a pravého syna (nejprve se aktualizují rodiče listových uzlů, pak jejich rodiče...). Poté se ověří, zda aktualizovaný strom je stále Huffmanův strom a to tak, že se pro každý uzel ověří, zda některý z uzlů, který je na stejné úrovni více vpravo nebo na vyšší úrovni (blíže ke kořenu), nemá nižší hodnotu jak daný uzel. Pokud ano, tak se uzly vymění, aktualizují se hodnoty čítačů předků vyměněných uzlů a spustí se znovu ověřování. Pokud se zkontrolují všechny uzly stromu, aniž by se některé prohodily, je daný strom Huffmanův a proces změny měřítka končí.

2.5 Přidání nového uzlu

Při přidání nového uzlu **char** reprezentujícího přečtený symbol se nejprve vytvoří **noChar** uzel, který se vloží do stromu místo uzlu **escape**. Uzel **Escape** se přidá jako levý syn nově vytvořeného **noChar** uzlu a poté se vytvoří **char** uzel, který se přidá jako pravý syn **noChar** uzlu.

2.6 Kód uzlu

Kód uzlu se získá jako cesta od kořene k danému uzlu, je-li uzel v cestě levým synem svého rodiče, zapisuje se na dané pozici v cestě nulový bit, je-li pravým synem svého rodiče, zapisuje se jedničkový bit.