KKO

Projekt 2

-

Komprese textových souborů s využitím Burrows-Wheelerovy transformace

Petr Kubát, xkubat11

Zadání

V jazyce C/C++ implementujte knihovnu a aplikaci pro kompresi a dekompresi textových souborů s využitím BWT transformace (Burrows-Wheeler).

Rozbor algoritmu

Implementace byla provedena v jazyce C++. Vlastní algoritmus je rozdělen do několika fází.
Komprese:

1. Ze vstupního souboru se načte blok dat (defaultně 10kB), přidá se speciální znak a pomocí suffix array se vstupní text zakóduje Burrows Wheelerovou transformací.
2. Na zakódovaný vstupní text se použije algoritmus Move to Front, který je vhodný pro použití s algoritmy pro entropické kódování.
3. V třetí fázi se aplikuje kódování Run Length Encoding, které pole bytů zakóduje následujícím způsobem:
   AABCCCCCDEEE -> AA0BCC3DEE1
4. V poslední fázi by mělo bý použito entroické kódování, avšak v této implementaci se pouze provádí výstup do souboru, přičemž je nutno do souboru přidat i oddělovač bloků a informaci o indexu do pole suffixů pro dekódování.

Dekomprese - algoritmus je v podstatě inverzní ke kompresi:

1. Ze vstupního souboru se načte index do pole suffixů a poté se čte, dokud program nenarazí na oddělovač bloků (char s hodnotou 255). Poté provede dekódování RLE zakódovaného slova.
2. Následně se provede dekódování pomocí Move to Front a BWT, opět pomocí suffix array. Z konce řetězce se vyjme znak vložený ve fázi komprese a provede se výstup do souboru.

Nedostaky

Z důvodu časové tísně nebylo implementováno entropické kódování, avšak aplikace je i tak schopna zkomprimovat vstupní testovací soubor o cca. 30KB (s defaultní velikostí bloku).
Vyšší komprese by mohlo být dosaženo vyjmutím kódováním Move to Front z aplikace, jelikož jeho použití bez entropického kódování je spíše přítěží.
Zároveň také časová náročnost komprimovacího algoritmu roste exponenciálně s velikostí bloku. Zapříčiněno nejspíše použitím C++ stringu.

Závěr

Zadání se podařilo splnit až na zmíněné nedostatky.