Codifica di Huffman
TDII

Marco Zucca <s4828628@studenti.unige.it>

12 maggio 2021

1 Informazioni preliminari

Il codice fornito é stato compilato con:

```
g++ src/btree.cpp src/huffman.cpp -o huffman
```

su macchina Linux x86_64 e gcc ver. 10.3.1 20210422 .
si puo' compilare con il parametro -D DEBUG per far stampare gli stati intermedi del processo di codifica

2 Funzionamento

2.1 File di output .info

Nel file .info viene inserito il numero di bit utilizzati per comprimere il file,la codifica utilizzata per comprimere il file nel formato: carattere:codifica|carattere:codifica|... l'entropia, la lumnghezza attesa e le lunghezze (in byte e in bit) del file dato in input e il relativo file compresso

2.2 File di output .huff

Nel file con estensione .huff viene salvato il file di input compresso con la codifica di huffman basata sulla frequenze dei caratteri del file in input.
Questo approccio richiede che insieme al file binario contenente il file compresso, sia anche fornito al decodificatore, la codifica utilizzata per la compressione (vedi .info). Salvare la compressione ottenuta su un file porta a delle complicazioni, perché non si puo' scrivere bit per bit la codifica sul file, in quanto la minima unitá scrivibile su un file é un byte. Per ovviare a cio' infondo al file vengono aggiunti dei bit di "padding" per poter raggiungere un byte (se necessario) e viene comunicato insieme al file contenente la codifica utilizzata, il numero di bit effettivi utilizzati per la codifica.

3 Osservazioni

3.1 Entropia e lunghezza attesa

Utlizzando come file di input testi con numero caratteri maggiori di 10^5 (vedi `input/dante.txt`) si nota che la lunghezza attesa calcolata sulla compressione é sempre maggiore rispetto all'entropia calcolata sul testo di input, anche se se di poco.
Questo accade perché l'entropia rappresenta il limite inferiore della lunghezza attesa di una codifica univocamente decifrabile (come quella di Huffman). Questa proprietá é data dal teorema che afferma, appunto, che la lunghezza attesa di una codifica univocamente decifrabile, non puo' essere minore dell'entropia.

3.2 Compressione

3.2.1 Rapporto tra input e output

Prendendo sempre come esempio `input/dante.txt`, é possibile vedere una netta riduzione dei bit utilizzati per rappresentare il file di input:

```
Number of char before:  557042(4456336 bit)
Number of char after:   322059(2576469 bit ,with padding 2576472)
```

Possiamo valutare quanti bit in meno sono stati calcolando:

$$G = \frac{after\ bits}{before\ bits} = \frac{2576469}{4456336} \approx 0.5781 \approx 57\%$$