Design document

Jussi Mäki - joamaki@gmail.com

Data structures project

Helsinki September 13, 2008

1 Introduction

This document describes the design of data structures project involving the use of the Huffman-coding algorithm.

The Huffman code algorithm is a greedy algorithm that takes a list of characters and corresponding frequencies in input data and outputs a tree which describes the optimal prefix code for each character. This resulting prefix code tree can be then used to compress the input data and to decompress it.

The program implemented in this project shall provide compression and decompression of arbitary data using the Huffman code algorithm.

2 Data structures

The huffman code algorithm as described in [1] requires the use of an auxilliary data structure to extract the lowest frequency object, ie. a form priority queue. A binary heap data structure is well suited for this task and so shall be used as the auxilliary data structure for the algorithm.

To build the set of character frequencies in the input data a table of integers the size of possible character values is used. This table is then rebuilt as a heap (using frequency as the key) before passing it to the huffman code algorithm.

3 Program usage

The program is controlled by the use of command-line arguments. It has two modes of operation: compression and decompression.

If no command-line arguments are given it'll default to compressing the standard input to the standard output. If one filename is given it will compress and save the output to filename.hc. Decompression mode is enabled with the option '-d'. If a filename is given when decompression mode is enabled and it has the suffix .hc the result is output to a file with the suffix removed.

Example of compression and decompression:

```
$ hc thesis.txt
Compressing thesis.txt to thesis.txt.hc...
$ hc -d thesis.txt.hc
Decompressing thesis.txt.hc to thesis.txt...
```

References

[1] Thomas H. Cormen. *Introduction to Algorithms.* MIT Press, 2001. ISBN-13: 978-0262032933.