Huffman

Theis Erik Jendal, stud no. 20157126, tjenda15@student.aau.dk

November 20, 2018

When running the program and calling the main function, one can enter a text and get the a bit encoding back. The user will then be prompted to either press one to decode the text again or to exit the method. If the user chooses to decode, there will be printed the decoded text, matching the input text.

I use functions defined in prelude, such as getLine, uncurry, compare and foldr. Further more i use the Data.List module, where I use sort and insert, which can be used as BTree (the datatype I have implented) is in both the Eq and Ord classes, where compares on the frequency of a node. The program works by finding the unique characters and their values and constructing a leaf node with these values. The Huffman tree is then constructed given a list of BTree's, it then uses the algorithm described in the book, more or less, to construct the tree. I use the Huffman tree to make a list of characters and their bit encoding, such that i can look up the character and get the bit encoding. Encoding the message is then easy using folding, append and lookup.

The encode function return both the bit encoding and the Huffman tree. The decode function can traverse the Huffman tree, by either going left or right in the tree dependant upon the next bit in the input sequence, or using cons a character to the rest of the decoded string if it is a leaf recursively. If a leaf node is reached, the rest of the recursive sequence have to have the root of the tree again.

I have defined two types and two synonyms. The first datatype Bit represents bits, with the term constructors Zero and One. The second datatype BTree is a binary tree with a term constructor Leaf that takes a Char, then a Int and return BTree ($Leaf :: Char -> Int -> BTree$)) and the other is $Branch :: BTree -> BTree -> Int -> BTree$. The BTree datatype is therefore able to represent the Huffman tree. Both datatypes are a part of the Show class. Furthermore BTree is also a part of Eq and Ord, where I have made a specific instance insure that BTrees are compared on their frequency.