CS 350 2019-20 Homework 4

Due Date November 14 2019

---

1. Permutation of a list : write a function which takes two inputs : the first, a list xs of n elements, the second - a permutation of the numbers 0,1,...,n-1. It should output the permuted list where the first element in xs goes to the position indicated by the first element in the permutation, the second element in xs goes to the position indicated by the second element in the permutation, etc.

   e.g.

   ```
   permute ['a','b','c']  [0, 2, 1]
   ```

   should result in ['a','c','b'].

   [10 points]

2. Write a haskell function compose which takes two arguments: the first argument is a list of functions, i.e. with type [a -> a]. The second must be a single element a. You have to do the following: if there are n functions $f_1, \ldots, f_n$, and the second argument is a then return $f_n(f_{n-1}(\ldots(f_1(a))\ldots))$.

   [10 points]

3. Define a function altMap :: [a] -> (a->b) -> (a->b) -> [b] that applies the first function on all elements in the first list which occur at odd positions, and applies the second function on all elements in the first list which occur at even positions.

   For example, altMap (+ 1) (* 2) [10,2,30,4,50] results in [11,4,31,8,51]. **[10 points]**

1. Implement a data type for Binary Trees. Your data type must be an instance of Eq, Show, Read, Applicative and Monad. **[25 points]**

2. The data type Ordering is defined as data Ordering = LT | EQ | GT. The standard function compare has type Ord a => a -> a -> Ordering. That is, given two values, it returns LT if the first value is less than the second, EQ if they are equal and GT otherwise.

   Implement a function occurs : Ord a => a -> BinaryTree a -> Bool which searches for a key within a Binary *search* tree represented using your binary tree representation. **[10 points]**

3. A directed graph is represented as a list of tuples (u, v) which represent an edge $u \rightarrow v$ in the digraph. Write a function which prints a single edge in the dot language notation. Use this and fmap to print the dot notation corresponding to the digraph. **[10 points]**
4. Write a haskell program to read a file "test.txt", and count the number of words in the file. You may assume that no word has been split across multiple lines. You may use library functions like getLine, words etc. to implement this function. Write your code a. using the do notation b. without using the do notation, and just using the >>= operator. **[10 points]**

5. Consider the functions

   ```
   safeFirst lst
    | (length lst)==0 = error "Empty List"
    | otherwise       = head lst
   ```

   and

   ```
   safeReciprocal x
    | x==0 = error "Undefined"
    | otherwise = 1/x
   ```

   These function is well-typed, and has type [a]->a and (Eq a, Fractional a) => a->Fractional a, respectively. With this understanding, explain how the error function can be implemented. **[10 points]**

Author: Satyadev Nandakumar
Created: 2019-11-07 Thu 21:12
Validate