CENG 242

Hw #2

Spring 2007/2008

(Due: March 30th, 2008 Sunday 23:59)

In this homework, you will first implement these two functions:

```
substring :: String -> String -> Bool
subsequence :: String -> String -> Bool
```

substring function will check whether the string given as second parameter is the substring of the first. For example:

```
substring “abcde” “bcd” ← True
substring “abcde” “bd” ← False
```

subsequence function will check whether the string given as second parameter is the subsequence of the first. For example:

```
subsequence “abcde” “bd” ← True
subsequence “abcde” “db” ← False
```

Detailed explanation of substring and subsequence can be found on wikipedia.

Then, you will implement a higher order function:

```
stringsearch :: [String] -> String -> (String -> String ->
Bool) -> [[String]]
```

This function will get a list of strings, a string and a function as parameters. Then it will check all the possible permutations of string list (first parameter of stringsearch) and add a permutation to the list to be returned if the given function returns true with the parameters, concatenation of the strings in the permutation and the second parameter of stringsearch. Let’s clarify it with an example:

```
stringsearch [“abc”,“bc”,“acc”] “bca” substring
```

First you will generate all the permutations of the first list:

[] , [“abc”] , [“bc”] , [“acc”] , [“abc”,“bc”] , [“abc”,“acc”] , [“bc”,“abc”] , [“bc”,“acc”] , [“acc”,“abc”] , [“acc”,“bc”] , [“abc”,“bc”,“acc”] , [“abc”,“acc”,“bc”] , [“bc”,“abc”,“acc”] , [“bc”,“acc”,“abc”] , [“acc”,“abc”,“bc”] , [“acc”,“bc”,“abc”]

Then, for all of these permutations, check whether “bca” is a substring of the concatenation the strings of the permutation. So, in this example, stringsearch function will return the following list:

[[“abc”,“acc”] , [“bc”,“abc”] , [“bc”,“acc”] , [“abc”,“bc”,“acc”] , [“abc”,“acc”,“bc”] , [“bc”,“abc”,“acc”] , [“bc”,“acc”,“abc”] , [“acc”,“bc”,“abc”]]

Specifications:

- All the work should be done **individually**.
- Your codes should be written in **Haskell** and have the name “**Hw2.hs”.**
- Your code should have the module **Hw2** (i.e. your code should start with: **module Hw2 where**)
- In evaluation, black box method will be used. So, be careful about names, types etc.
- You will submit your codes through **cow** system.
- You should test your codes in **inek** machines with **hugs** before submitting.