CMPS 258 – PROGRAMMING LANGUAGES – SPRING 2021

WEEK 3 ASSIGNMENT

In this assignment, you will write a higher-order function for traversing the file system object datatype from the previous assignment. You will then re-implement some of the functions you implemented in the previous assignment using the higher-order function with anonymous functions as arguments. The datatype binding is shown below:

```
datatype FSObject = File      of string * int
                  | Link      of string * string
                  | Directory of string * (FSObject list)
```

1. Write a higher-order function `traverseFS` that takes a file system object and a function for each variant of the `FSObject` datatype (and a base case for the list variant), and traverses the datatype by applying the right function for each variant. (Hint: Look at your functions from the previous assignment and see what they have in common and where they differ.)
2. Use `traverseFS` to re-implement `totalSize` from the previous assignment. Pass anonymous functions to `traverseFS`.
3. Use `traverseFS` to re-implement `containsLinks` from the previous assignment. Pass anonymous functions to `traverseFS`.
4. Use `traverseFS` to re-implement `getFilesLargerThan` from the previous assignment. Pass anonymous functions to `traverseFS`.
5. Use `traverseFS` to re-implement `countLinksTo` from the previous assignment. Pass anonymous functions to `traverseFS`.
6. This question is unrelated to the previous questions. The following function `concatStrings` takes a `string list` and returns a `string` that is the concatenation of all the strings in the list in order from beginning to end:
   ```
   fun concatStrings xs =
       case xs of
           []     => ""
         | x::xs' => x ^ concatStrings(xs')
   ```
   Rewrite this function using tail recursion.

Evaluating a correct homework solution should generate the bindings below. However, keep in mind that generating these bindings does not guarantee that your solutions are correct. Make sure to test your functions before submitting.

```
datatype FSObject
  = Directory of string * FSObject list
  | File of string * int
  | Link of string * string
val traverseFS = fn
  : FSObject * (string * int -> 'a) * (string * string -> 'a) * 'a *
    ('a * 'a -> 'a)
    -> 'a
val totalSize = fn : FSObject -> int
val containsLinks = fn : FSObject -> bool
val getFilesLargerThan = fn : FSObject * int -> string list
val countLinksTo = fn : FSObject * string -> int
val concatStrings = fn : string list -> string
```

Assessment

Solutions should be:

- Correct
- In good style, including indentation and line breaks
- Written using features discussed in class

Submission Instructions

Put all your solutions in one SML file and submit it via Moodle. The file should be named "<id>.sml" where <id> is your AUBnet ID (e.g., abc01.sml). Do not submit any other files or compressed folders.