DM552 exercises

Department of Mathematics and Computer Science
University of Southern Denmark

September 20, 2017

1. Under what conditions on xs and ys does the following equation hold?

$$[x \mid x \leftarrow xs, y \leftarrow ys] \equiv [x \mid y \leftarrow ys, x \leftarrow xs]$$

2. Define appropriate versions of the library functions

$$\begin{array}{ll}
\mathit{repeat} & :: a \rightarrow [a] \\
\mathit{repeat}\ x & = xs\ \textbf{where}\ xs = x : xs \\
\mathit{take} & :: \mathit{Int} \rightarrow [a] \rightarrow [a] \\
\mathit{take}\ n\ _ \mid n \leqslant 0 & = [\,] \\
\mathit{take}\ _\ [\,] & = [\,] \\
\mathit{take}\ n\ (x : xs) & = x : \mathit{take}\ (n-1)\ xs \\
\mathit{replicate} & :: \mathit{Int} \rightarrow a \rightarrow [a] \\
\mathit{replicate}\ n & = \mathit{take}\ n \circ \mathit{repeat}
\end{array}$$

for the following type of binary trees:

$$\textbf{data}\ \mathit{Tree}\ a = \mathit{Nil} \mid \mathit{Node}\ a\ (\mathit{Tree}\ a)\ (\mathit{Tree}\ a)$$

3. You are given the following data types:

$$\begin{array}{l}
\textbf{data}\ \mathit{Direction} = L \mid R\ \textbf{deriving}\ \mathit{Show} \\
\textbf{data}\ \mathit{Tree}\ a = \mathit{Nil} \mid \mathit{Node}\ a\ (\mathit{Tree}\ a)\ (\mathit{Tree}\ a)\ \textbf{deriving}\ \mathit{Show}
\end{array}$$

The direction data type will be used to define a path in the tree. Define the following functions:

$$\mathit{elementAt} :: [\mathit{Direction}] \rightarrow \mathit{Tree}\ a \rightarrow \mathit{Maybe}\ a$$

which - for a given path and tree - returns the value at the node reached, if the path is followed (it returns *Nothing* in case the path is invalid, or specifies a *Nil*-node).

$$modifyAt :: (a \to a) \to [\mathit{Direction}] \to \mathit{Tree}\ a \to \mathit{Tree}\ a$$

which - for a given path and tree - applies a function to the value at the node. Invalid paths will just leave the tree unchanged.

4. Define a function

$$toTree :: [a] \to \mathit{Tree}\ a$$

which converts a list to a tree, filling each level in the tree from left to right.

$[1, 2, 3, 4, 5]$ would be converted to

$$\mathit{Node}\ 1\ (\mathit{Node}\ 2\ (\mathit{Node}\ 4\ \mathit{Nil}\ \mathit{Nil})\ (\mathit{Node}\ 5\ \mathit{Nil}\ \mathit{Nil}))\ (\mathit{Node}\ 3\ \mathit{Nil}\ \mathit{Nil})$$

HINT: First define a function which splits a list into levels: $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$ becomes $[[1], [2, 3], [4, 5, 6, 7], [8, 9, 10]]$. Then define left- and right descends on this level list representation of the tree. Using these functions, one can make the conversion to the *Tree*-datatype.

5. You are given types to represent expressions in a small interpreted language.

```
data Expression =
   Var String   -- Variable
   | Val Int    -- Integer literal
   | Op Expression Bop Expression   -- Operation
   deriving (Show, Eq)

   -- Binary (2-input) operators
data Bop =
   Plus
   | Minus
   | Times
   | Divide
   | Gt
   | Ge
   | Lt
   | Le
```

$$\begin{aligned} &\mid \mathit{Eql} \\ &\mathbf{deriving}\ (\mathit{Show}, \mathit{Eq}) \end{aligned}$$

We store the currently defined variables in a function (see the last slides from lecture 3)

$$\mathbf{type}\ \mathit{State} = \mathit{String} \rightarrow \mathit{Maybe}\ \mathit{Int}$$

Implement an evaluator for expressions:

$$\mathit{evalE} :: \mathit{State} \rightarrow \mathit{Expression} \rightarrow \mathit{Either}\ \mathit{String}\ \mathit{Int}$$

The evaluator either returns an error message or the evaluated value - for boolean outputs, the convention is that 0 is false and 1 is true.