Homework 2 — ML core language — assigned Saturday 18 February — due Sunday 26 February

Reading assignment

Read Chapter 4 of *ML for the Working Programmer*.

General instructions

Carefully state all preconditions for all function arguments. Make sure that your tests cover the space of permissible argument values well.

You will find yourself writing numerous recursive functions over lists. For each such function, carefully consider and document (in code comments) its traversal pattern. For each such function, determine if it can be expressed as an application of some built-in (typically higher-order) function from the `List` structure in the Standard ML Basis Library. If so, comment out the original recursive definition and provide a new definition in terms of library functions.

2.1 Using lists for arithmetic: writing recursive functions over lists (50pts) [A.3; E; K.2.2]

Numerals can be represented as lists of integers. For instance, decimal numerals can be expressed as lists of integers from 0 to 9. The integer 12345678901234567890 might be represented as the ML list `[1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0]: int list`. However, the representation should allow a radix other than 10 as well.

We use the following declaration:

```
datatype numeral = Numeral of {radix: int, digits: int list}
```

The above example number is represented as:

```
val example =
   Numeral {radix = 10,
            digits = [1, 2, 3, 4, 5, 6, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0]}
```

Write the following functions:

1. (10pts) `makeLongInt: int -> int -> numeral`, such that `makeLongInt` n r computes the list representation of the integer n in radix r. You can assume that $n \geq 0$, and that $r > 1$. For example,
   `makeLongInt 123 10`
   should evaluate to
   `Numeral {radix=10, digits=[1,2,3]}: numeral`.

2. (10pts) `evaluateLongInt: numeral -> int`, such that
   `evaluateLongInt (Numeral {radix=`r`, digits=`l`})`

converts a numeral back to an ML integer. You can assume that *l* is a valid list for radix *r*, and the numerical value of the list is small enough to fit into an ML `int`. For example,
`evaluateLongInt (Numeral {radix=10, digits=[1,2,3]})`
should evaluate to
`123: int`.

3. (10pts) `changeRadixLongInt: numeral -> int -> numeral`, such that
`changeRadixLongInt` *N r*
computes the representation of the same number as *n* in a new radix *r*. For example,
`changeRadixLongInt (Numeral {radix=10, digits=[1,2,3]}) 8`
should evaluate to
`Numeral {radix=8, digits=[1,7,3]}: numeral`.

4. (10pts) `addLongInts: numeral * numeral -> numeral`, such that
`addLongInts (`*a*`,`*b*`)`
computes the sum of the numbers given by the numerals *a* and *b*. If *a* and *b* use the same radix, that radix should be used for the result. If *a* and *b* use different radices, the result should use the larger one. For example,
`addLongInts (Numeral {radix=10, digits=[1,2,3]}, Numeral {radix=3, digits=[1]})`
should evaluate to
`Numeral {radix=10, digits=[1,2,4]}: numeral`.
Important: the arguments *a* and *b* can represent arbitrarily large integers, unlike ML's built-in `int` type.

5. (10pts) `mulLongInts: numeral * numeral -> numeral`, such that
`mulLongInts (`*a*`,`*b*`)`
computes the product of the numbers given by the numerals *a* and *b*. If *a* and *b* use the same radix, that radix should be used for the result. If *a* and *b* use different radices, the result should use the larger one. For example,
`mulLongInts (Numeral {radix=10, digits=[1,2,3]}, Numeral {radix=3, digits=[1]})`
should evaluate to
`Numeral {radix=10, digits=[1,2,3]}: numeral`.
Important: the arguments *a* and *b* can represent arbitrarily large integers, unlike ML's built-in `int` type.

2.2 Modelling (30pts) [A.1; K.2.2]

This is a continuation of Homework Problem 1.6. In that problem a fluidic network was shown. It consisted of successive stages. Each stage did a splitting and mixing, characterized by the number of inlets and the number of outlets, such that the number of outlets of a stage was always one more that the number of its inlets. Furthermore, successive stages in the network had successively larger numbers of inlets. Consider a generalization of the fluidic mixing network from that problem. Successive stages can have arbitrary numbers of inlets and outlets.

We describe a stage as:

```
datatype stage = Stage of {inlets: int, outlets: int}
```

We describe a network as:

```
datatype network = Network of stage list
```

For example, the six-stage network from Figure 1 in Problem 1.6 is:

```
Network
   [
    Stage {inlets=3, outlets=4},
    Stage {inlets=4, outlets=5},
    Stage {inlets=5, outlets=6},
    Stage {inlets=6, outlets=7},
    Stage {inlets=7, outlets=8},
    Stage {inlets=8, outlets=9}
   ]
```

2.2.1 Sanity check (10pts)

Write a function `networkOK: network -> bool` which checks that the description is consistent in the sense that the number of outlets of a stage is equal to the number of inlets of the next stage.

2.2.2 Stage mixing (10pts)

Develop the transfer matrix $\mathbf{M}_{s,r}$ for a stage with r inlets and s outlets.

Declare

```
type realmatrix = real list list
```

where the representation of matrices is either row-major or column-major (but state which one you have chosen).

Write a function `mixstage: stage -> realmatrix` such that `mixstage (Stage {inlets=`r`, outlets=`s`})` computes $\mathbf{M}_{s,r}$.

2.2.3 Network mixing (10pts)

Write a function `mixnetwork: network -> realmatrix` which computes the transfer matrix for an entire mixing network (precondition: the network passes the sanity check).

2.3 Matrix inverse (20pts)

This is another continuation of Homework Problem 1.6. Write a function `minv: realmatrix -> realmatrix` to compute the matrix inverse.

How to turn in

Turn in your code by running

~marron/cs451/handin your-file

on a regular UNM CS machine. You should use whatever filename is appropriate in place of your-file. Whole directories may be submitted as well.

Include the following statement with your submission, signed and dated:
I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents' Policy Manual.