Programming Languages

Homework 2

Due 2:20 pm, April 8, 2009

1. Read Chapter 2 (The Core Language; pp. 17–42) of *Notes on Programming Standard ML of New Jersey*, by Riccardo Pucella. The pdf file is available at
`http://www.cs.cornell.edu/riccardo/prog-smlnj/notes-011001.pdf`

2. (5 points) This assignement asks you to implement the denotational semantics of While programs in Standard ML, following the description in Section 4.3.3 in the text book but with some change. Please use the ML datatypes defined below in your implementation:

```
datatype iexpr = Int     of int
               | Var     of string
               | Plus    of iexpr * iexpr
               | Minus   of iexpr * iexpr
               | Times   of iexpr * iexpr
               | Divide  of iexpr * iexpr

datatype bexpr = Bool      of bool
               | And       of bexpr * bexpr
               | Or        of bexpr * bexpr
               | Not       of bexpr
               | Eq        of iexpr * iexpr
               | Less      of iexpr * iexpr
               | LessEq    of iexpr * iexpr
               | Greater   of iexpr * iexpr
               | GreaterEq of iexpr * iexpr

datatype prog = Empty
              | Assignment  of string * iexpr
              | Sequence    of prog * prog
              | Conditional of bexpr * prog * prog
              | While       of bexpr * prog
```

Note that we represent expresions and programs in ML using values of datatypes `iexpr`, `bexpr`, and `prog`. Datatype `iexpr` is for integer expressions, `bexpr` is for boolean expressions, and `prog` is for While programs (note that empty programs are allowed).

The original While program below (p. 71, text book)

```
x:= 0; y:= 0; while x <= z do (y := y+x; x:= x+1)
```

is now represented by the following value `ex_4_5` in ML

```
val ex_4_5 = Sequence (Assignment ("x", Int 0),
                       Sequence (Assignment ("y", Int 0),
                                 While (LessEq (Var "x", Var "z"),
                                        Sequence (Assignment ("y", Plus (Var "y", Var "x")),
                                                  Assignment ("x", Plus (Var "x", Int 1))))))
```

For states, they are represented in ML as functions of type `string -> int`. The denotational semantics of While programs is now a function of type `prog -> (string -> int) -> string -> int`.

(a) Implement function `modify` in ML such that `modify s x a` will return a new state that is just like `s`, except that the value of `x` is `a`. Function `modify` will be used to implement the semantics of assignment. Note that we only allow integer variables in states.

(b) Complete the following ML code fragment (`....` marks the places you need to fill in)

```
fun e_i (Int   i)    s = i
  | e_i (Var   v)    s =       ....
  |                  ....

fun e_b (Bool  b)    s = b
  | e_b (And (u, v)) s =     .....
  |                  ....

fun c  Empty         s =  ....
  |                  ....
```

so that functions `e_i`, `e_b`, and `c` give out the semantics of integer expressions, boolean expressions, and While programs as intended. In particular, by your implementation, the following ML program

```
fun all0 x = 0
val s0 = modify all0 "z" 2

val s = c ex_4_5 s0

val x = s "x"
val y = s "y"
val z = s "z"
```

will produce the following results

```
val x = 3 : int
val y = 3 : int
val z = 2 : int
```

(c) Write an interesting While program of your own (using datatype `prog`, of course). Explain what your program does, and use the denotational semantics above to run several interesting test cases.

3. (5 points) Read Section 4.3.4 in the text book. You are asked to implement in ML a nonstandard semantics of While program such that it analyzes programs to see if all variables in the program are initialized before they are used. That is, you are asked to implement the example in pp. 74–76.

You can reuse the ML code above, or you can use your own ML definitions for expressions, programs, and states.

1 PLEASE NOTE, NO EXCEPTION

- Homework is due **before the class begins** on April 8. Late homework will not be accepted.
- For this programming assignment, you must hand in **printout of the code, as well as the testing data and result**. Programs must be accompanied by their documentations.
- You are expected to do the homework by yourself. Discussion among peers is encouraged but **copying from others is a shame**.