T-501-FMAL Programming languages, Practice class 11
Spring 2021

All of the following problems are about MicroC.

1. Provide an example of an accessor `a` for which the pre-increment expression `++a` is not equivalent to the assignment expression `a = a + 1`.

2. Render `++a` equivalently via `=` and `+` nonetheless, using a call to a function encapsulating the behavior of `++`.

3. Consider the following modification to the case of `=` (assignment) of expression evaluation in the interpreter of MicroC (in MicroC.fs).

   ```
   let rec eval e locEnv gloEnv sto : int * store =
       match e with
       ...
       | Assign (acc, e) ->
            let loc, sto' = access acc locEnv gloEnv sto
            let res, sto'' = eval e locEnv gloEnv sto'
            getSto sto' loc, setSto sto'' loc res       // the change is in this line
       ...
   ```

   What is the effect of modification, as compared to the usual semantics of `=` in (Micro)C?

   In particular, what will the following program print? Explain.

   ```
   int x;
   int y;
   int z;
   x = 3;
   y = 4;
   z = 5;
   x = y = z;       // = associates to the right, ie x = (y = z) is meant
   printi x;
   printi y;
   printi z;
   ```

4. What does the following program print? Explain.

   (You can either reason based on your knowledge of the semantics of MicroC or render the program in abstract syntax and run the interpreter.)

   ```
   int* p;
   int* q;
   int x;
   x = 3;
   p = &x;
   printi *p;
   q = p;
   *p = *q * 5;
   printi x;
   ```

5. What does the following program print? Explain.

   ```
   int* p;
   int* q;
   int x;
   int y;
   x = 5;
   ```

```
y = 7;
p = &x;
q = &y;
printi *q;
*p = *q - 3;
printi x;
```