Comp 317: Semantics of Programming Languages

Problem Sheet 7

1. Simplify the following (check your answers using Maude):

   (a) `[[ 'x := 2 ; if ('y < 'x){ 'y := 'y + 1; } else { skip } ]](initial)`

   (b) `[[ 'x := 2 ; while ('y < 'x){ 'y := 'y + 1 ; 'z := 'z + 'y ; } ]](initial)`

   (c) `[[ 'z ]]([[ while ('y < 2){ 'y := 'y + 1 ; 'z := 'z + 'y ; } ]](initial))`

2. Prove that the following program swaps the values of `'x` and `'y`.

   ```
   'x := 'x + 'y ; 'y := 'x - 'y ; 'x := 'x - 'y ;
   ```

3. Write a program that sets `'a` to the maximum of the values of `'x` and `'y`. Prove that your program is correct.

4. Extend the Maude syntax and semantics of SIMPLE with case-conditionals.

5. (Tricky!) Extend the Maude syntax and semantics of SIMPLE with post-increments: expressions of the form `V++`, which as expressions give the value of the variable `V`, but also have the side-effect of incrementing that value. Comparing this with the Class Test, you will want to make use of pairs `< I , S >`, where `I` is an integer, and `S` is a Store. Such pairs can be specified in Maude as follows.

   ```
   sort  IntStorePair .
   op  <_,_>  : Int Store -> IntStorePair .
   op  getInt  : IntStorePair -> Int .
   op  getStore  : IntStorePair -> Store .
   var  I : Int .
   var  S : Store .
   eq  getInt(< I , S >)  =  I .
   eq  getStore(< I , S >)  =  S .
   ```