COMP 317: Semantics of Programming Languages

Problem Sheet 2

UNIVERSITY OF LIVERPOOL

Attempt these questions *before* your tutorial session.

---

1. Here's something you could have seen on the whiteboard during lectures: ([[`'x := 25;`]](*S*))(`'x`). Take a couple of minutes to look at it.
   1. Make sense of it, and explain it.
   2. Simplify it.

2. Write a program that swaps the values of the variables `'x` and `'y`. I.e., after running your program, the value stored in the variable `'x` should be the value that was previously stored in the variable `'y`, and the value stored in the variable `'y` should be the value that was previously stored in the variable `'x`.
   Hint: use a "temporary storage" variable `'t`.

   Prove that your program does what it's meant to do.
   Hint: simplify [[*your program here*]](*S*) for an arbitrary state *S*.

   Now prove that `'x := 'y; 'y := 'x;` *doesn't* swap the values of `'x` and `'y`.
   Hint: simplify [[`'x := 'y; 'y := 'x;`]](*S*) for an arbitrary state *S*.

3. Write a program that sets `'z` to either the value of `'a` or the value of `'b`, whichever is the greater.

   Test your program by simplifying

   - [[`'a := -2; 'b := 4;` *your program here*]](*S*), and
   - [[`'a := 9; 'b := 4;` *your program here*]](*S*).

   Challenge: why is this "test" and not "prove"?
   Mega-challenge: how would you prove and not test?

4. Write a program (using a while-loop) that sets `'f` to 100!. Note that *n*! (pronounced *n factorial*) is the product 1 * 2 * 3 * ... * *n*.

   Now prove your program is correct by simplifying
   [[*your program here*]](*S*).
   - Only kidding! replace `100` in your program by `3` and simplify
   [[*your program with 100 replaced by 3 here*]](*S*), and you should see that `'f` is set to 1*2*3 = 6.

   Challenge: we'll see later on in this module how to prove your program is correct, but right now can you explain why it's correct?

5. Stare at this program until you understand it:

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

   Now prove that it does what you think it does.

Grant Malcolm