Lab 3

The following exercises require you to use the Dafny definitions provided in the lecture notes (including `AExp`, `BExp`, `Stmt`, `State`, `evalAExp`, `evalBExp`, and `evalStmt`).

Exercises

1. (0.5p) Write a lemma to assert the result of the evaluation for `(x + 5) * (y + -3)` in state σ=`{x↦2,y↦5}`.

2. (0.5p) Write a lemma to assert the result of `!(x < 4) && (y < (x + y))`, where the state is σ=`{x↦5,y↦10}`.

3. (1p) Write a lemma that uses the `evalStmt` predicate to prove the final state σf=`{x↦10,y↦100}` is obtained after executing the sequence `x := 10; y := x * x`, where the initial state is σ=`{x↦0,y↦0}`.

4. (1p) Write a lemma that uses the `evalStmt` predicate to prove that the evaluation of `if (x < y) then x := y + 1 else skip` in a state σ=`{x↦5,y↦8}` is taking the `then` branch and the final state is σf=`{x↦9,y↦8}`.

   > *Hint: The Assign statement requires g=1 and the If statement requires g equal to the gas of the chosen branch.*

5. (1p) Write a lemma to prove that a while loop `while (0 < x) do x := x + 1` terminates immediately (zero iterations) in a state σ=`{x↦0}`.

6. (1p) Consider the following Dafny code:

```
lemma ex6()
{
var assign1 := Assign(x, Num(15));
var assign2 := Assign(y, Num(15));
var assign3 := Assign(z, Var(y));
var assign4 := Assign(z, Var(x));
var cond := Less(Var(x), Var(y));
var iff := If(cond, assign3, assign4);
var seq1 := Seq(assign1, assign2);
var seq2 := Seq(seq1, iff);

var sigma := map[x := 0, y := 0];
var sigma1 := sigma[x := 15];
var sigma2 := sigma1[y := 15];
var sigma3 := sigma2[z := 15];

// fill in here some helper assertions for Dafny
assert(evalStmt(seq2, sigma, sigma3,3));
}
```

   The final assertion does not hold. Add some intermediate assertions to help Dafny find the proof for the final assertion `assert(evalStmt(seq2, sigma, sigma3,3));`.

Last updated: 11/06/2025 09:13:26 - Copyright © Andrei Arusoaie