COSC 4780 Name: Jacob Tuttle
Principles of Programming Langauges

Homework 05

March 11, 2020

Apologies for the weird margins, I had troubles fitting the long equations onto the page otherwise.

4.6:

$$< x := y; y := 3; x := x + 3, s > \to s'$$
$$[COMP_{BSS}] :< x := y, s > \to s'' < y := 3; x := x + 3, s'' > \to s'$$
$$[ASS_{BSS}] : s[x \mapsto 4] \to s'' < y := 3; x := x + 3, s'' > \to s'$$
$$[COMP_{BSS}] :< y := 3, s'' > \to s^{(3)} < x := x + 3, s^{(3)} \to s'$$
$$[ASS_{BSS}] : s''[y \mapsto 3] \to s^{(3)} < x := x + 3, s^{(3)} > \to s$$
$$[VAR_{BSS}] :< x := 4 + 3, s^{(3)} > \to s'$$
$$[ASS_{BSS}] : s'[x \mapsto 7, y \mapsto 3]$$

4.7:

$$< i := 1; while(x \neq 0)do(i := i * x; x := x - 1), s > \to s'$$
$$[COMP_{BSS} :] < i := i, s > \to s'' < while(x \neq 0)do(i := i * x; x := x - 1), s'' > \to s'$$
$$[ASS_{BSS} :]s[i \mapsto 1, x \mapsto 3] \to s'' < while(x \neq 0)do(i := i * x; x := x - 1), s'' > \to s'$$
$$[WHILE - T_{BSS} :] < i := i * x; x := x - 1, s'' > \to s^{(3)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(3)} > \to s'$$
$$[COMP_{BSS} :] < i := i * x, s'' > \to s^{(4)} < x := x - 1, s^{(4)} > \to s^{(3)}$$
$$[VAR_{BSS}x2 :] < i := 1 * 3, s'' > \to s^{(4)} < x := x - 1, s^{(4)} > \to s^{(3)}$$
$$[ASS_{BSS} :]s''[i \mapsto 3, x \mapsto 3] \to s^{(4)} < x := x - 1, s^{(4)} > \to s^{(3)}$$
$$[ASS_{BSS} :]s^{(4)}[i \mapsto 3, x \mapsto 2] \to s^{(3)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(3)} > \to s'$$
$$[WHILE - T_{BSS} :] < i := i * x; x := x - 1, s^{(5)} > \to s^{(6)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(6)} > \to s'$$
$$[COMP_{BSS} :] < i := i * x, s^{(5)} > \to s^{(7)} < x := x - 1, s^{(7)} > \to s^{(6)}$$
$$[VAR_{BSS}x2 :] < i := 3 * 2, s^{(5)} > \to s^{(7)} < x := x - 1, s^{(7)} > \to s^{(6)}$$
$$[ASS_{BSS} :]s^{(5)}[i \mapsto 6, x \mapsto 2] \to s^{(7)} < x := x - 1, s^{(7)} > \to s^{(6)}$$
$$[ASS_{BSS} :]s^{(7)}[i \mapsto 6, x \mapsto 1] \to s^{(6)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(6)} > \to s'$$
$$[WHILE - T_{BSS} :] < i := i * x; x := x - 1, s^{(8)} > \to s^{(9)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(9)} > \to s'$$
$$[COMP_{BSS} :] < i := i * x, s^{(8)} > \to s^{(10)} < x := x - 1, s^{(10)} > \to s^{(9)}$$
$$[VAR_{BSS}x2 :] < i := 6 * 1, s^{(8)} > \to s^{(10)} < x := x - 1, s^{(10)} > \to s^{(9)}$$
$$[ASS_{BSS} :]s^{(8)}[i \mapsto 6, x \mapsto 1] \to s^{(10)} < x := x - 1, s^{(10)} > \to s^{(9)}$$
$$[ASS_{BSS} :]s^{(10)}[i \mapsto 6, x \mapsto 0] \to s^{(9)} < while(x \neq 0)do(i := i * x; x := x - 1), s^{(9)} > \to s'$$
$$[WHILE - F_{BSS} :] < while(x \neq 0)do(i := i * x; x := x - 1), s^{(9)} > \to s'$$
$$s'[i \mapsto 6, x \mapsto 0]$$

4.17: Since big-step semantics has only one transition sequence, we don't need to show something being true for all possible length k transitions.

4.18: Prove, using a suitable proof technique, that the big-step semantics of statements is *deterministic*, that is, that for any statement S and state s we have that if $< S, s > \rightarrow s'$ and $< S, s > \rightarrow s''$ then $s' = s''$. (You may assume that the big-step semantics of arithmetic and Boolean expressions are deterministic.)

5.13: $\sim$ means that two statements are semantically equivalent. Since semantic equivalence is the formal idea of "same behavior" and we know that `abort` and `skip` don't have the same behavior (the first ends the program without returning a state while the second proceeds to execute following commands or just returns the current state) we know that the two actions are not similar in either big or small-step semantics.

5.14: Since neither `while 0=0 do skip` nor `abort` return a state, they both have the same behavior and thus are semantically equivalent.