Program Syntax; Operational Semantics

CS 536: Science of Programming, Fall 2022

A. Why

- Our simple programming language is a model for the kind of constructs seen in actual languages.
- Step-by-step program evaluation can be described using a sequence of program / state snapshots.

B. Outcomes

At the end of today, you should be able to

- Read and write simple programs in our programming language.
- Translate simple programs in our language to and from C / C++ / Java.
- Describe the step-by-step execution of a program in our language by giving its operational semantics.

C. Problems

Part I: Program Syntax

1. In our simple language, ***if*** *x < 0* ***then*** *x := 0* ***fi*** is (syntactically) equivalent to what other statement?
2. How are ***if*** B ***then*** S_1 ***else*** S_2 ***fi*** and ***if*** B ***then*** e_1 ***else*** e_2 ***fi*** different?

For Questions 3 – 8, translate the given C / C++ / Java program fragments into our simple programming language.

3. `++x; if (x < y) { x = y = y+1; }`
4. `y = z * ++x; z = z+x;`
5. `y = z * x++; z = z+x;`
6. `x = z = 0; while (x++ < n) z = z+x;`
7. `z = 1; for (x = n ; x >= 1 ; --x) z = z * x;`
8. `x = 0; while (x++ <= n) { y = (++x)*y;}`

Part II: Operational Semantics

9. Evaluate each of the following configurations to completion. If there are multiple steps, show each step individually.
   a. $\langle x := x+1, \{x = 5\}\rangle$
   b. $\langle y := 2*x, \{x = 6\}\rangle$
   c. $\langle x := x+1, \sigma\rangle$ (Your answer will be symbolic — you'll need to include $\sigma(x)$.)
   d. $\langle x := x+1;\ y := 2*x, \{x = 5\}\rangle$

10. Let $S \equiv$ ***if*** $x > 0$ ***then*** $x := x+1$ ***else*** $y := 2*x$ ***fi***.
    a. Let $\sigma(x) = 8$, evaluate $\langle S, \sigma\rangle$ to completion, showing the individual steps. Give the final state.
    b. Repeat, if $\sigma(x) = 0$.
    c. Repeat, if we don't know what $\sigma(x)$ is. (Your answer will be symbolic.)

11. Let $S \equiv$ ***if*** $x > 0$ ***then*** $x := x/z$ ***fi***. Evaluate S (starting) in σ, for each the σ below:
    a. $\sigma = \{x = 8, z = 3\}$ (and don't forget, integer division truncates)
    b. $\sigma = \{x = -2, z = 3\}$

12. Let $W \equiv$ ***while*** $x < 3$ ***do*** S ***od*** where $S \equiv x := x+1;\ y := y*x$.
    a. Show what evaluation of the body S in an arbitrary state τ does.
    b. Use your answer from part a to evaluate W in σ where $\sigma \vDash x = 4 \wedge y = 1$.
    c. Repeat part b where $\sigma \vDash x = 1 \wedge y = 1$.

CS 536: Solution to Practice 5 (Program Syntax; Operational Semantics)

Part I: Syntax

1. ***if*** *x < 0* ***then*** *x := 0* ***else skip fi***
2. ***if*** *B* ***then*** S_1 ***else*** S_2 ***fi*** is a statement; its evaluation can change the state.
   if *B* ***then*** e_1 ***else*** e_2 ***fi*** is an expression; its evaluation produces a value.
3. *x:=x+1;* ***if*** *x < y* ***then*** *y:=y+1; x:=y* ***fi***
4. *x:=x+1; y:=z*x; z:=z+x*
5. *y:=z*x; x:=x+1; z:=z+x*
6. *z:=0; x:=z;* ***while*** *x < n* ***do*** *x:=x+1; z:=z+x* ***od****; x:=x+1*
7. *z:=1; x:=n;* ***while*** *x >= 1* ***do*** *z:=z*x; x:=x-1* ***od***
8. In the solution below, the increment of *x* after the ***od*** is for the *x++* of the test that breaks out of the loop. For the body of the loop, the first increment of *x* is for the *x++* in *x++ <= n* after testing *x <= n*. The immediately following increment of *x* is for the *++x* in *y = (++x)*y* because the increment occurs before calculating *y = x*y*. You could certainly combine the two *x := x+1* to just one *x := x+2*.

   x := 0; ***while*** *x<= n* ***do*** *x := x+1; x := x+1; y := x*y* ***od;*** *x := x+1*

Part II: Operational Semantics

9. (Calculate meanings of programs)

a. $\langle x := x+1, \{x = 5\}\rangle \to \langle E, \tau\rangle$ where $\tau = \{x = 5\}[x \mapsto \{x = 5\}(x+1)]$
   $= \{x = 5\}[x \mapsto 6] = \{x = 6\}$.

b. $\langle y := 2*x, \{x = 6\}\rangle \to \langle E, \tau\rangle$ where $\tau = \{x = 6\}[y \mapsto \{x = 6\}(2*x)]$
   $= \{x = 6\}[y \mapsto 12] = \{x = 6, y = 12\}$

c. $\langle x := x+1, \sigma\rangle \to \langle E, \sigma[x \mapsto \sigma(x+1)]\rangle = \langle E, \sigma[x \mapsto \sigma(x)+1]\rangle$

d. $\langle x := x+1; y := 2*x, \{x = 5\}\rangle$
   $\to \langle y := 2*x, \{x = 5\}[x \mapsto \alpha]\rangle$ *where* $\alpha = \{x = 5\}(x+1) = 6$
   $= \langle y := 2*x, \{x = 5\}[x \mapsto 6]\rangle$
   $= \langle y := 2*x, \{x = 6\}\rangle$
   $\to \langle E, \{x = 6\}[y \mapsto \beta]\rangle$ *where* $\beta = \{x = 6\}(2*x) = 12$
   $= \langle E, \{x = 6, y = 12\}\rangle$

10. Let $S \equiv$ ***if*** *x > 0* ***then*** *x := x+1* ***else*** *y := 2*x* ***fi***.

a. If $\sigma(x) = 8$, then $\sigma(x > 0) = T$,
   so $\langle S, \sigma\rangle \to \langle x := x+1, \sigma\rangle \to \langle E, \sigma[x \mapsto \sigma(x+1)]\rangle = \langle E, \sigma[x \mapsto 9]\rangle$.

b. If $\sigma(x) = 0$, then $\sigma(x > 0) = F$,
   so $\langle S, \sigma\rangle \to \langle y := 2*x, \sigma\rangle \to \langle E, \sigma[y \mapsto \sigma(2*x)]\rangle = \langle E, \sigma[y \mapsto 0]\rangle$

c. If $\sigma(x) > 0$ then $\langle S, \sigma\rangle \to \langle x := x+1, \sigma\rangle = \langle E, \sigma[x \mapsto \sigma(x)+1]\rangle$.
If $\sigma(x) \le 0$ then $\langle S, \sigma\rangle \to \langle y := 2*x, \sigma\rangle = \langle E, \sigma[y \mapsto 2 \times \sigma(x)]\rangle$.

11. Let $S \equiv$ ***if*** $x > 0$ ***then*** $x := x/z$ ***fi*** $\equiv$ ***if*** $x > 0$ ***then*** $x := x/z$ ***else skip fi***

a. If $\sigma = \{x = 8, z = 3\}$, then $\sigma(x > 0) = T$ so $\langle S, \sigma\rangle \to \langle x := x/z, \sigma\rangle \to \langle E, \sigma[x \mapsto \alpha]\rangle$ where $\alpha = \sigma(x/z) = \sigma[x \mapsto 8/3] = \sigma[x \mapsto 2]$, since integer division truncates.

b. If $\sigma = \{x = -2, z = 3\}$ then $\sigma(x > 0) = F$, so $\langle S, \sigma\rangle \to \langle$***skip***$, \sigma\rangle \to \langle E, \sigma\rangle$.

12. Let $W \equiv$ ***while*** $x < 3$ ***do*** S ***od*** where $S \equiv x := x+1; y := y*x$.

a. For arbitrary τ, $\langle S,\tau\rangle \to \langle x := x+1; y := y*x,\tau\rangle \to \langle y := y*x,\tau[x \mapsto \tau(x)+1]\rangle$
$\to \langle E, \tau[x \mapsto \tau(x)+1][y \mapsto \alpha]\rangle$ where $\alpha = \tau[x \mapsto \tau(x)+1](y*x) = \tau(y) \times (\tau(x)+1)$.

b. If $\sigma \models x = 4 \wedge y = 1$, then $\sigma(x < 3) = F$ so $\langle W, \sigma\rangle \to \langle E, \sigma\rangle$.

c. If $\sigma \models x = 1 \wedge y = 1$, then $\sigma(x < 3) = T$ so we have at least one iteration to do.
Let $\sigma_0 = \sigma$, let $\sigma_1 = \sigma_0(y) \times (\sigma_0(x)+1)$, and let $\sigma_2 = \sigma_1(y) \times (\sigma_1(x)+1)$. Then

$\sigma_0 = \sigma[x \mapsto 1][y \mapsto 1]$

$\sigma_1 = \sigma_0[x \mapsto \sigma_0(x)+1][y \mapsto \sigma_0(y) \times (\sigma_0(x)+1)] = \sigma[x \mapsto 2][y \mapsto 2]$

$\sigma_2 = \sigma_1[x \mapsto 2+1][y \mapsto 2 \times (2+1)] = \sigma[x \mapsto 3][y \mapsto 6]$

Since σ_0 and $\sigma_1 \models x < 3$ but $\sigma_2 \models x \ge 3$, we have
$\langle W, \sigma\rangle \to \langle S; W, \sigma_0\rangle \to^* \langle W, \sigma_1\rangle = \langle S; W, \sigma_1\rangle \to^* \langle W, \sigma_2\rangle \to \langle E, \sigma_2\rangle$, so σ_2 is the final state.

