IIT CS440: Programming Languages and Translators

Homework 3: Big-step semantics and Lambda calculus

Out: Wednesday, Mar. 30
Due: Sunday, Apr 10, 11:59pm

0 Logistics and Submission - Important

Please submit your solutions as a PDF (typed or *neatly* handwritten!) on Blackboard by the due date.

1 Big Step Semantics

The following are the big-step semantic rules of the IMP language, whose syntax was described in class.

$$\frac{}{< i, \sigma >\Downarrow_e i}\ \text{(CONST, WHEN I IS AN INTEGER)} \qquad \frac{}{< u, \sigma >\Downarrow_e v}\ \text{(VAR, IF U:=V} \in \sigma\text{)}$$

$$\frac{< e_1, \sigma >\Downarrow_e v_1 \qquad < e_2, \sigma >\Downarrow_e v_2}{< e_1 \oplus e_2, \sigma >\Downarrow_e v_1 \oplus v_2}\ \text{(ARITH)}$$

Figure 1: Big-step expression rules.

$$\frac{}{< i, \sigma >\Downarrow_b b}\ \text{(CONST, WHEN B IS TRUE OR FALSE)} \qquad \frac{}{< u, \sigma >\Downarrow_b v}\ \text{(VAR, IF U:=V} \in \sigma\text{)}$$

$$\frac{< e_1, \sigma >\Downarrow_e v_1 \qquad < e_2, \sigma >\Downarrow_e v_2}{< e_1 \backsim e_2, \sigma >\Downarrow_b v_1 \backsim v_2}\ \text{(COMP)}$$

Figure 2: Big-step boolean rules.

$$\frac{}{< \textbf{skip}, \sigma >\Downarrow \sigma} \text{ (SKIP)} \qquad \frac{< e, \sigma >\Downarrow_e v}{< x := e, \sigma >\Downarrow \sigma[x := v]} \text{ (ASSIGN)}$$

$$\frac{< S_1, \sigma >\Downarrow \sigma' \qquad < S_2, \sigma' >\Downarrow \sigma''}{< S_1; S_2, \sigma >\Downarrow \sigma''} \text{ (SEQ)} \qquad \frac{< B, \sigma >\Downarrow_b \textbf{true} \qquad < S_1, \sigma >\Downarrow \sigma'}{< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2, \sigma >\Downarrow \sigma'} \text{ (IF1)}$$

$$\frac{< B, \sigma >\Downarrow_b \textbf{false} \qquad < S_2, \sigma >\Downarrow \sigma'}{< \texttt{if } B \texttt{ then } S_1 \texttt{ else } S_2, \sigma >\Downarrow \sigma'} \text{ (IF2)} \qquad \frac{< B, \sigma >\Downarrow_b \textbf{false}}{< \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma >\Downarrow \sigma} \text{ (WHILE1)}$$

$$\frac{< B, \sigma >\Downarrow_b \textbf{true} \qquad < S, \sigma >\Downarrow \sigma' \qquad < \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma' >\Downarrow \sigma''}{< \textbf{while } B \textbf{ do } S \textbf{ od}, \sigma >\Downarrow \sigma''} \text{ (WHILE2)}$$

Figure 3: Big-step statement rules.

For each of the following assertions, write a big-step proof tree. An example proof tree for:

$$< m := x + 1, \{x := 3\} >\Downarrow \{x := 3, m := 4\}$$

is given below:

$$\dfrac{\dfrac{\dfrac{}{< x, \{x := 3\} >\Downarrow_e 3} \text{ (VAR)} \qquad \dfrac{}{< 1, \{x := 3\} >\Downarrow_e 1} \text{ (CONST)}}{< x + 1, \{x := 3\} >\Downarrow_e 4} \text{ (ARITH)}}{< m := x + 1, \{x := 3\} >\Downarrow \{x := 3, m := 4\}} \text{ (ASSIGN)}$$

Task 1.1 (Written, 25 points).

(a) $< x, \{x := 20\} >\Downarrow_e 20$

(b) $< x || \textbf{false}, \{x := \textbf{true}\} >\Downarrow_b \{\textbf{true}\}$

(c) $< t := a + b; a := b; b := t, \{a := 5, b := 10\} >\Downarrow \{t := 15, a := 10, b := 15\}$

(d) $< \textbf{if } x > y \textbf{ then } m := x * 10 \textbf{ else } m := y * 10, \{x := 10, y := 20\} >\Downarrow \{x := 10, y := 20, m := 200\}$

(e) $< \textbf{while } x < 1 \textbf{ do } x := x + 1 \textbf{ od}, \{x := 0\} >\Downarrow \{x := 1\}$

2 λ Calculus

For each of the following pairs of λ-terms, say whether or not they are equivalent. If they are, give the steps with what kind of equivalence/reduction you're using. For example, if the question is

$$(\lambda x.\lambda y.x\ y)\ \lambda z.z \stackrel{?}{\equiv} \lambda u.(\lambda v.(\lambda z.z)\ v)\ u$$

you could write

$$(\lambda x.\lambda y.x\ y)\ \lambda z.z \equiv_\beta \lambda y.(\lambda z.z)\ y \equiv_\alpha \lambda v.(\lambda z.z)\ v \equiv_\eta \lambda u.(\lambda v.(\lambda z.z)\ v)\ u$$

Task 2.1 (Written, 25 points).

(a) $x \stackrel{?}{\equiv} z$

(b) $(\lambda y.\lambda z.y)\ x \stackrel{?}{\equiv} \lambda y.x$

(c) $x \stackrel{?}{\equiv} \lambda x.x\ x$

(d) $\lambda y.x \stackrel{?}{\equiv} (\lambda z.\lambda y.x)\ z$

(e) $\lambda y.x \stackrel{?}{\equiv} \lambda z.(\lambda y.x)\ z$