C311 October 2018: Exercise 2

Neil Ghani and Conor McBride

October 18, 2018

For each of the following, mark each of the claims 1 for True and 0 for False in the associated Marx-file. While you may collaborate within your designated groups, you should submit your own answer. Please submit your answer by Monday 29 October 12pm. Refer to the lecture notes for the various semantics for IMP and LAM that we have covered in the lecures and which will be used here.

Part A Consider IMP-expression e given by $\texttt{new}\ x := y+2\ \texttt{in do}\ x := x-1\ \texttt{return}\ x+y$. Which of the following claims are true?

1. If σ is $, y := 4$ then $\sigma \,|\, e \Downarrow, y := 4, x := 5 \,|\, 9$
2. If σ is $, y := 4, z := 5$ then $\sigma \,|\, e \Downarrow, y := 4, z := 5 \,|\, 9$
3. If σ is $, y := 4, x := 5$ then $\sigma \,|\, e \Downarrow, y := 4 \,|\, 9$
4. If σ is $, z := 5$ then $\sigma \,|\, e \Downarrow, z := 5 \,|$

Part B: Consider the IMP-expression e given by

$$\texttt{new}\ x := 8\ \texttt{in}\ (\texttt{new}\ x := 6\ \texttt{in do}\ x := x + y\ \texttt{return}\ x) + x$$

. Which of the following claims are true?

1. If σ is $, y := 4$ then $\sigma \,|\, e \Downarrow, y := 5, x := 8 \,|\, 18$
2. If σ is $, y := 5, x := 3$ then $\sigma \,|\, e \Downarrow, y := 5 \,|\, 15$
3. If σ is $, y := 15, x := 3$ then $\sigma \,|\, e \Downarrow, y := 5, x := 3 \,|\, 29$
4. If σ is $, y := 15$ then $\sigma \,|\, e \Downarrow, y := 5 \,|\, 29$

Part C: Consider the LAM-term *two* given by $\backslash f \to \backslash x \to f(fx)$ and the lambda term *id* given by $\backslash x \to x$. Using the small step reduction semantics for LAM, which of the following claims are true?

1. There are no infinite reduction sequences starting from *two id*
2. All reduction sequences from *two two id* eventually end up at *id*
3. The term *two* $(x\ y)$ reduces to $\backslash x \to (x\ y)((x\ y)\ x)$
4. The term *two two* reduces to $\backslash f \to \backslash x \to f(f(f(fx)))$

Part D: Using the big-step, call-by-name, reduction semantics for LAM, and the terms *two* and *id* as defined above, which of the following claims are true?

1. *two id* $\Downarrow \backslash x \to$ *id* (*id* x)
2. *two two id* $\Downarrow$ *two id*

3. $(3 + two) \Downarrow 5$
4. $two\ (\backslash x \to (x + x))\ 7 \Downarrow 28$

Part E: Using the big-step environment semantics for LAM, which of the following claims are true

1. $, x := 3 \,|\, (\backslash x \to (x + x))\ x \Downarrow, x := 3 \,|\, 12$
2. $, x := 3 \,|\, (\backslash x \to (x + x))\ id \Downarrow [, x := 3](id + id)$
3. $, x := 3, y := 6 \,|\, (\backslash x \to (x + y))\ 5 \Downarrow [, x = 3, y := 6]11$
4. $, x := 3, y := []z \to z \,|\, (\backslash x \to y\ x)\ 5 \Downarrow 5$

Part F: We plan to devise a *small step* semantics for IMP, and we need your help. We have the following judgement forms, following the same metavariable naming conventions as in the lecture notes.

- $\sigma_0|e_0 \rightsquigarrow \sigma_1|e_1$
  starting with store σ_0, integer expression e_0 steps to store σ_1 and integer expression e_1
- $\sigma_0|p_0 \rightsquigarrow \sigma_1|p_1$
  starting with store σ_0, boolean expression p_0 steps to store σ_1 and boolean expression p_1
- $\sigma_0|c_0 \rightsquigarrow \sigma_1|c_1$
  starting with store σ_0, executing command c steps to store σ_1 and command c_1
- $\sigma_0|cs_0 \rightsquigarrow \sigma_1|cs_1$
  starting with store σ_0, executing block cs steps to store σ_1 and block cs_1

Now, remembering that every integer value is an integer expression, and the same for booleans, the idea is to compute one step at a time until the expression evolves to a value. The command that lays the role of a 'command value' is $\{\}$, the command which gives nothing left to do. The *empty* block is the 'block value'. Of course, by the time we reach such an empty value, all the changes made by the command or the block will be in the new store.

The small step semantics should agree with our big step semantics in the following ways:

- For integer expressions, exactly when $\sigma \,|\, e \Downarrow \sigma' \,|\, v$, there should be a (possibly empty) sequence of small steps $\sigma|e \rightsquigarrow \sigma_1|e_1\ \ \sigma_1|e_1 \rightsquigarrow \sigma_2|e_2\ \ \ldots\ \ \sigma_n|e_n \rightsquigarrow \sigma'|v$
- For boolean expressions, exactly when $\sigma \,|\, p \Downarrow \sigma' \,|\, b$, there should be a (possibly empty) sequence of small steps $\sigma|p \rightsquigarrow \sigma_1|p_1\ \ \sigma_1|p_1 \rightsquigarrow \sigma_2|p_2\ \ \ldots\ \ \sigma_n|p_n \rightsquigarrow \sigma'|b$
- For commands, exactly when $\sigma \,|\, c \Downarrow \sigma'$, there should be a (possibly empty) sequence of small steps $\sigma|c \rightsquigarrow \sigma_1|c_1\ \ \sigma_1|c_1 \rightsquigarrow \sigma_2|c_2\ \ \ldots\ \ \sigma_n|c_n \rightsquigarrow \sigma'|\{\}$
- For blocks, exactly when $\sigma \,|\, cs \Downarrow \sigma'$, there should be a (possibly empty) sequence of small steps $\sigma|cs \rightsquigarrow \sigma_1|cs_1\ \ \sigma_1|cs_1 \rightsquigarrow \sigma_2|cs_2\ \ \ldots\ \ \sigma_n|cs_n \rightsquigarrow \sigma'|$

We are confident in the following rules:

$$\frac{\sigma|e \rightsquigarrow \sigma'|e'}{\sigma|x := e \rightsquigarrow \sigma'|x := e'} \qquad \frac{}{\sigma, x := v, x\backslash\sigma'|x := v' \rightsquigarrow \sigma, x := v', \sigma'|\{\}}$$

$$\frac{\sigma|cs \rightsquigarrow \sigma'|cs'}{\sigma|\{cs\} \rightsquigarrow \sigma'|\{cs'\}}$$

$$\frac{}{\sigma|\texttt{if}\ (0)\ c_1\ \texttt{else}\ c_0 \rightsquigarrow \sigma|c_0}$$

$$\frac{\sigma|e_0 \rightsquigarrow \sigma\}e_0'|}{\sigma|\texttt{new}\ x := e_0\ \texttt{in}\ e_1 \rightsquigarrow \sigma'|\texttt{new}\ x := e_0'\ \texttt{in}\ e_1} \qquad \frac{}{\sigma|v_0\texttt{+}v_1 \rightsquigarrow \sigma|v_0 + v_1}$$

Note that, in the rule for addition, the step is from the addition of two values to the value which is their sum: it is not doing nothing!

For each of the following rules, we claim that they are good rules to add if we want to achieve the correspondence we seek. We want you to tell us if, in each case, the claim is true.

1. $$\frac{}{\sigma|\{\}; cs \ \leadsto \ \sigma|cs}$$

2. $$\frac{\sigma|c \ \leadsto \ \sigma'|c'}{\sigma|c; cs \ \leadsto \ \sigma'|c'; cs}$$

3. $$\frac{\sigma|cs \ \leadsto \ \sigma'|cs'}{\sigma|c; cs \ \leadsto \ \sigma'|c; cs'}$$

4. $$\frac{\sigma|c_1 \ \leadsto \ \sigma'|c_1'}{\sigma|\texttt{if } (p) \ c_1 \texttt{ else } c_0 \ \leadsto \ \sigma'|\texttt{if } (p) \ c_1' \texttt{ else } c_0}$$

5. $$\frac{}{\sigma|\texttt{while } (p) \ c \ \leadsto \ \sigma|\{c; \texttt{if } (p) \ \{\texttt{while } (p) \ c; \} \texttt{ else } \{\}\}}$$

6. $$\frac{}{\sigma|\texttt{while } (p) \ c \ \leadsto \ \sigma|\texttt{if } (p) \ \{c; \texttt{while } (p) \ c; \} \texttt{ else } \{\}}$$

7. $$\frac{}{\sigma, x := v, x \backslash \sigma'|x \ \leadsto \ \sigma|v}$$

8. $$\frac{\sigma|e_0 \ \leadsto \ \sigma'|e_0'}{\sigma|e_0 \texttt{+} e_1 \ \leadsto \ \sigma'|e_0' \texttt{+} e_1}$$

9. $$\frac{\sigma|e_1 \ \leadsto \ \sigma'|e_1'}{\sigma|e_0 \texttt{+} e_1 \ \leadsto \ \sigma'|e_0 \texttt{+} e_1'}$$

10. $$\frac{\sigma|e_1 \ \leadsto \ \sigma'|e_1'}{\sigma|v_0 \texttt{+} e_1 \ \leadsto \ \sigma'|v_0 \texttt{+} e_1'}$$

11. $$\frac{\sigma, x := e_0|e_1 \ \leadsto \ \sigma', x := e_0'|e_1'}{\sigma|\texttt{new } x := e_0 \texttt{ in } e_1 \ \leadsto \ \sigma'|\texttt{new } x := e_0' \texttt{ in } e_1'}$$

12. $$\frac{}{\sigma|\texttt{new } x := n \texttt{ in } n' \ \leadsto \ \sigma|n'}$$

Of course, there are still plenty of rules missing. We do not claim we have the full set of small step rules: e.g., we have given no rules for boolean expressions, so far. It may help to think about what they should be, and try some examples.