Harvard School of Engineering and Applied Sciences — CS 152: Programming Languages

IMP; Denotational Semantics
Section and Practice Problems

Section Week 3

---

1 IMP

Consider the small-step operational semantics for IMP given in Lecture 5. Let σ_0 be a store that maps all program variables to zero.

(a) Find a configuration $\langle c, \sigma' \rangle$ such that $\langle \textbf{if } 8 < 6 \textbf{ then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0 \rangle \longrightarrow \langle c, \sigma' \rangle$ and give a derivation showing that $\langle \textbf{if } 8 < 6 \textbf{ then } \mathsf{foo} := 2 \textbf{ else } \mathsf{bar} := 8, \sigma_0 \rangle \longrightarrow \langle c, \sigma' \rangle$.

(b) What is the sequence of configurations that

$$\langle \mathsf{foo} := \mathsf{bar} + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0 \rangle$$

steps to? (You don't need to show the derivations for each step, just show what configuration $\langle \mathsf{foo} := \mathsf{bar} + 3; \textbf{if } \mathsf{foo} < \mathsf{bar} \textbf{ then skip else } \mathsf{bar} := 1, \sigma_0 \rangle$ steps to in one step, then two steps, then three steps, and so on, until you reach a final configuration.)

Now consider the large-step operational semantics for IMP given in Lecture 5. Let σ_0 be a store that maps all program variables to zero.

(c) Find a store σ' such that $\langle \textbf{while } \mathsf{foo} < 3 \textbf{ do } \mathsf{foo} := \mathsf{foo} + 2, \sigma_0 \rangle \Downarrow \sigma'$ and give a derivation showing that $\langle \textbf{while } \mathsf{foo} < 3 \textbf{ do } \mathsf{foo} := \mathsf{foo} + 2, \sigma_0 \rangle \Downarrow \sigma'$.

(d) Suppose we extend boolean expressions with negation.

$$b ::= \cdots \mid \textbf{not } b$$

(i) Give an inference rule or inference rules that show the (large step) evaluation of **not** b.

2 Denotational Semantics

(a) Give the denotational semantic for each of the following IMP programs. That is, express the meaning of each of the following programs as a function from stores to stores.

(i) $\mathsf{a} := \mathsf{b} + 5; \mathsf{a} := \mathsf{a} \times \mathsf{b}$

(ii) $\textbf{if } \mathsf{foo} < 0 \textbf{ then } \mathsf{bar} := \mathsf{foo} \times \mathsf{foo} \textbf{ else } \mathsf{bar} := \mathsf{foo} \times \mathsf{foo} \times \mathsf{foo}$

(iii) $\mathsf{bar} := \mathsf{foo} \times \mathsf{foo}; \textbf{if } \mathsf{foo} < 0 \textbf{ then skip else } \mathsf{bar} := \mathsf{bar} \times \mathsf{foo}$
(Hint: the answer to this question should be the same function as the answer to Question 2(a).ii above. You may have written the function down differently, but it should be the same mathematical function.)

(iv) $\mathsf{a} := 0; \mathsf{b} = 0; \textbf{while } \mathsf{a} < 3 \textbf{ do } \mathsf{b} := \mathsf{b} + \mathsf{c}$

(b) Consider the following loop.

$$\textbf{while } \mathsf{foo} < 5 \textbf{ do } \mathsf{foo} := \mathsf{foo} + 1; \mathsf{bar} := \mathsf{bar} + 1$$

We will consider the denotational semantics of this loop.

(i) What is the denotational semantics of the loop guard $\mathsf{foo} < 5$? That is, what is the function $\mathcal{B}[\![\mathsf{foo} < 5]\!]$?

(ii) What is the denotational semantics of the loop body $\mathsf{foo} := \mathsf{foo} + 1; \mathsf{bar} := \mathsf{bar} + 1$? That is, what is the function $\mathcal{C}[\![\mathsf{foo} := \mathsf{foo} + 1; \mathsf{bar} := \mathsf{bar} + 1]\!]$?

(iii) Recall that the semantics of the loop is the fixed point of the following higher-order function F. (This is from Section 1.2 of Lecture 6, where we have provided a specific loop guard b and loop body c for the higher-order function $F_{b,c}$).

$$\begin{aligned}
F &: (\textbf{Store} \rightharpoonup \textbf{Store}) \to (\textbf{Store} \rightharpoonup \textbf{Store}) \\
F(f) &= \{(\sigma, \sigma) \mid (\sigma, \textsf{\textbf{false}}) \in \mathcal{B}[\![\mathsf{foo} < 5]\!]\} \;\cup \\
&\qquad \{(\sigma, \sigma') \mid (\sigma, \textsf{\textbf{true}}) \in \mathcal{B}[\![\mathsf{foo} < 5]\!] \wedge \\
&\qquad\qquad \exists \sigma''.\ ((\sigma, \sigma'') \in \mathcal{C}[\![\mathsf{foo} := \mathsf{foo} + 1; \mathsf{bar} := \mathsf{bar} + 1]\!] \wedge (\sigma'', \sigma') \in f)\}
\end{aligned}$$

That is, the semantics of the loop are:

$$\begin{aligned}
\mathcal{C}[\![\textsf{\textbf{while}}\ \mathsf{foo} < 5\ \textsf{\textbf{do}}\ \mathsf{foo} := \mathsf{foo} + 1; \mathsf{bar} := \mathsf{bar} + 1]\!] &= \bigcup_{i \geq 0} F^i(\emptyset) \\
&= \emptyset \cup F(\emptyset) \cup F(F(\emptyset)) \cup F(F(F(\emptyset))) \cup \ldots \\
&= \mathrm{fix}(F)
\end{aligned}$$

Compute $F(\emptyset)$, $F(F(\emptyset))$, and $F(F(F(\emptyset)))$.
In general, what is the domain of the partial function $F^i(\emptyset)$? (Note that $F^i(\emptyset)$ is F applied to the empty set i times, e.g., $F^3(\emptyset)$ is $F(F(F(\emptyset)))$.)