

2010 Paper 6 Question 9

Semantics of Programming Languages

A very simple imperative language, L0, has the following syntax and semantics.

Locations: l, l_1, l_2, \dots (infinite)
Syntax: $e ::= \text{true} \mid \text{false} \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid l := e \mid !l$
Store: finite partial functions s from locations to $\{\text{true}, \text{false}\}$
Configuration: pairs $\langle e, s \rangle$ of an expression e and a store s
Type: bool (this is the only type)
Environment: a finite set Γ of locations

$$\begin{array}{l}
 \text{(r-if1)} \quad \langle \text{if true then } e_1 \text{ else } e_2, s \rangle \longrightarrow \langle e_1, s \rangle \\
 \text{(r-if2)} \quad \langle \text{if false then } e_1 \text{ else } e_2, s \rangle \longrightarrow \langle e_2, s \rangle \\
 \text{(r-if3)} \quad \frac{\langle e, s \rangle \longrightarrow \langle e', s' \rangle}{\langle \text{if } e \text{ then } e_1 \text{ else } e_2, s \rangle \longrightarrow \langle \text{if } e' \text{ then } e_1 \text{ else } e_2, s' \rangle} \\
 \text{(r-deref)} \quad \langle !l, s \rangle \longrightarrow \langle b, s \rangle \quad \text{if } l \in \text{dom}(s) \text{ and } s(l) = b \\
 \text{(r-assign1)} \quad \langle l := b, s \rangle \longrightarrow \langle b, s\{l \mapsto b\} \rangle \quad \text{if } l \in \text{dom}(s) \text{ and } b = \text{true or } b = \text{false} \\
 \text{(r-assign2)} \quad \frac{\langle e, s \rangle \longrightarrow \langle e', s' \rangle}{\langle l := e, s \rangle \longrightarrow \langle l := e', s' \rangle}
 \end{array}$$

$$\begin{array}{l}
 \text{(t-bool1)} \quad \Gamma \vdash \text{true} : \text{bool} \quad \text{(t-bool2)} \quad \Gamma \vdash \text{false} : \text{bool} \\
 \text{(t-deref)} \quad \Gamma \vdash !l : \text{bool} \quad \text{if } l \in \Gamma \quad \text{(t-assign)} \quad \frac{\Gamma \vdash e : \text{bool}}{\Gamma \vdash l := e : \text{bool}} \quad \text{if } l \in \Gamma \\
 \text{(t-if)} \quad \frac{\Gamma \vdash e : \text{bool} \quad \Gamma \vdash e_1 : \text{bool} \quad \Gamma \vdash e_2 : \text{bool}}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2 : \text{bool}}
 \end{array}$$

- (a) State the Progress theorem for well-typed L0. [2 marks]
- (b) Prove the Progress theorem, by rule induction on the structure of type derivations. [9 marks]
- (c) Define a notion of semantic equivalence for L0. Give a constraint on the syntax of e under which $(\text{if } e \text{ then } e_1 \text{ else } e_2)$ is semantically equivalent to (e_1) . [4 marks]
- (d) We now write $(e; e')$ as a shorthand for $(\text{if } e \text{ then } e' \text{ else } e')$. We say that two L0 expressions, e_1 and e_2 , form a “snap-back pair” if for every L0 expression e , the expression $((e_1; e); e_2)$ is semantically equivalent to (true) . Either exhibit a snap-back pair, or argue informally why there are no snap-back pairs in L0. [5 marks]