# 2010 Paper 6 Question 9

## Semantics of Programming Languages

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

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

$$\text{(r-if1)} \quad \langle \mathsf{if\ true\ then}\ e_1\ \mathsf{else}\ e_2, s\rangle \longrightarrow \langle e_1, s\rangle$$

$$\text{(r-if2)} \quad \langle \mathsf{if\ false\ then}\ e_1\ \mathsf{else}\ e_2, s\rangle \longrightarrow \langle e_2, s\rangle$$

$$\text{(r-if3)} \quad \frac{\langle e, s\rangle \longrightarrow \langle e', s'\rangle}{\langle \mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2, s\rangle \longrightarrow \langle \mathsf{if}\ e'\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2, s'\rangle}$$

$$\text{(r-deref)} \quad \langle !l, s\rangle \longrightarrow \langle b, s\rangle \quad \text{if } l \in \mathrm{dom}(s) \text{ and } s(l) = b$$

$$\text{(r-assign1)} \quad \langle l := b, s\rangle \longrightarrow \langle b, s\{l \mapsto b\}\rangle \quad \text{if } l \in \mathrm{dom}(s) \text{ and } b = \mathsf{true} \text{ or } b = \mathsf{false}$$

$$\text{(r-assign2)} \quad \frac{\langle e, s\rangle \longrightarrow \langle e', s'\rangle}{\langle l := e, s\rangle \longrightarrow \langle l := e', s'\rangle}$$

$$\text{(t-bool1)} \quad \Gamma \vdash \mathsf{true} : \mathsf{bool} \qquad \text{(t-bool2)} \quad \Gamma \vdash \mathsf{false} : \mathsf{bool}$$

$$\text{(t-deref)} \quad \Gamma \vdash !l : \mathsf{bool} \quad \text{if } l \in \Gamma \qquad \text{(t-assign)} \quad \frac{\Gamma \vdash e : \mathsf{bool}}{\Gamma \vdash l := e : \mathsf{bool}} \quad \text{if } l \in \Gamma$$

$$\text{(t-if)} \quad \frac{\Gamma \vdash e : \mathsf{bool} \quad \Gamma \vdash e_1 : \mathsf{bool} \quad \Gamma \vdash e_2 : \mathsf{bool}}{\Gamma \vdash \mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_2 : \mathsf{bool}}$$

(*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 ($\mathsf{if}\ e\ \mathsf{then}\ e_1\ \mathsf{else}\ e_1$) is semantically equivalent to ($e_1$). [4 marks]

(*d*) We now write $(e; e')$ as a shorthand for ($\mathsf{if}\ e\ \mathsf{then}\ e'\ \mathsf{else}\ e'$). 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 ($\mathsf{true}$). Either exhibit a snap-back pair, or argue informally why there are no snap-back pairs in L0. [5 marks]