## 9 Semantics of Programming Languages (pes20)

This mini-C language has mutable variables and block-structured scope:

*expression*, $e ::= \quad n \mid id \mid id = e \mid e\,;e' \mid \{\,\texttt{int}\,id_1\,;\,..\,\texttt{int}\,id_i\,;e\}$

Its operational semantics can be expressed in terms of environments $E$ that are partial functions mapping identifiers $id$ to the numeric addresses $n \in \mathbb{N}$ they are allocated at, memory heaps $H$ that are partial functions from addresses $\mathbb{N}$ to values $\mathbb{N}$, atomic evaluation contexts $A ::= \quad id = \text{-} \mid \text{-}\,;e'$, evaluation contexts $C ::= \quad \text{-} \mid C \cdot A$ which are lists of those, stacks $S ::= \quad \texttt{nil} \mid F :: S$ which are lists of a stack frame for each enclosing block, where a stack frame $F ::= \quad \langle C, E\rangle$ consists of an evaluation context and the environment for that block's local variables, and configurations $\langle e,\, S,\, H\rangle$. When we combine partial functions, e.g. with $H, H'$, their domains must be disjoint. Initial configurations are $\langle e,\, \langle \text{-}, \text{emp}\rangle :: \texttt{nil},\, \text{emp}\rangle$, writing emp for empty partial functions.

$$\frac{\begin{array}{l}\text{lookup } S\, id = n \\ H(n) = n'\end{array}}{\langle id, S, H\rangle \to \langle n', S, H\rangle}\ \text{VAR} \qquad \frac{\text{lookup } S\, id = n}{\langle id = n', S, (H, n \mapsto n_0)\rangle \to \langle n', S, (H, n \mapsto n')\rangle}\ \text{ASSIGN}$$

$$\frac{}{\langle n\,;e, S, H\rangle \to \langle e, S, H\rangle}\ \text{SEQ\_INT}$$

$$\frac{\begin{array}{c}E' = id_1 \mapsto n_1,\, ..\,, id_i \mapsto n_i \\ H' = n_1 \mapsto \texttt{0},\, ..\,, n_i \mapsto \texttt{0}\end{array}}{\langle \{\,\texttt{int}\,id_1\,;\,..\,\texttt{int}\,id_i\,;e\}\ , S, H\rangle \to \langle e, \langle \text{-}, E'\rangle :: S, (H, H')\rangle}\ \text{BLOCK\_START}$$

$$\frac{}{\langle n, (\langle \text{-}, E\rangle :: S), H\rangle \to \langle n, S, H \setminus \text{ran}\,(E)\rangle}\ \text{BLOCK\_END}$$

$$\frac{\neg\,(\text{isvalue}\,(e))}{\langle A[e], (\langle C, E\rangle :: S), H\rangle \to \langle e, (\langle C \cdot A, E\rangle :: S), H\rangle}\ \text{EVAL\_CTX\_FOCUS}$$

$$\frac{}{\langle n, (\langle C \cdot A, E\rangle :: S), H\rangle \to \langle A[n], (\langle C, E\rangle :: S), H\rangle}\ \text{EVAL\_CTX\_DEFOCUS}$$

(*a*) Define the lookup function. [4 marks]

(*b*) Give the transition sequence, with the configuration and rule name for each transition, of $\langle \{\,\texttt{int x;x = 1}\}\ \texttt{;y},\, \langle \text{-}, \text{emp}\rangle :: \texttt{nil},\, \text{emp}\rangle$. Include a brief explanation alongside each transition. [10 marks]

(*c*) Explain, with examples and reference to the rules, but without giving transition sequences, how this language treats variable shadowing. [4 marks]

(*d*) Explain what changes would be needed to add global variables. [2 marks]