2007 Paper 6 Question 10

Semantics of Programming Languages

Concurrent threads can interfere with each other by accessing the same store, so their behaviour can be nondeterministic and hard to reason about. This question develops a simple sufficient condition to rule that out, showing that two threads that do not share any store locations cannot interfere with each other's behaviour.

Consider the following mild variant of $\mathsf{L1}_{|}$, with distinguished grammars of expressions and processes, and corresponding reduction relations $\longrightarrow$ and $\twoheadrightarrow$.

Integers $n \in \mathbb{Z}$ Locations $\ell \in \mathbb{L} = \{l, l_0, l_1, l_2, \ldots\}$
Expressions $e ::= n \mid \mathbf{skip} \mid !\ell \mid \ell := e \mid e_1; e_2$ Processes $p ::= e \mid p_1 \mid p_2$
Stores s, finite partial functions from $\mathbb{L}$ to $\mathbb{Z}$

(e-deref) $\langle !\ell, s\rangle \longrightarrow \langle n, s\rangle$ if $\ell \in \mathrm{dom}(s)$ and $s(\ell) = n$

(e-assign1) $\langle \ell := n, s\rangle \longrightarrow \langle \mathbf{skip}, s + \{\ell \mapsto n\}\rangle$ if $\ell \in \mathrm{dom}(s)$

(e-assign2) $$\frac{\langle e, s\rangle \longrightarrow \langle e', s'\rangle}{\langle \ell := e, s\rangle \longrightarrow \langle \ell := e', s'\rangle}$$

(e-seq1) $\langle \mathbf{skip}; e_2, s\rangle \longrightarrow \langle e_2, s\rangle$

(e-seq2) $$\frac{\langle e_1, s\rangle \longrightarrow \langle e_1', s'\rangle}{\langle e_1; e_2, s\rangle \longrightarrow \langle e_1'; e_2, s'\rangle}$$

(p-thread) $$\frac{\langle e, s\rangle \longrightarrow \langle e', s'\rangle}{\langle e, s\rangle \twoheadrightarrow \langle e', s'\rangle}$$

(p-par1) $$\frac{\langle p_1, s\rangle \twoheadrightarrow \langle p_1', s'\rangle}{\langle p_1 | p_2, s\rangle \twoheadrightarrow \langle p_1' | p_2, s'\rangle}$$

(p-par2) $$\frac{\langle p_2, s\rangle \twoheadrightarrow \langle p_2', s'\rangle}{\langle p_1 | p_2, s\rangle \twoheadrightarrow \langle p_1 | p_2', s'\rangle}$$

Write $s \uplus s'$ for the union of two stores that have disjoint domain, and let $\mathrm{loc}(e)$ denote the set of store locations mentioned in e.

(a) Give a counterexample to [One-step determinacy for processes]:
If $\langle p, s\rangle \twoheadrightarrow \langle p_1, s_1\rangle$ and $\langle p, s\rangle \twoheadrightarrow \langle p_2, s_2\rangle$ then $\langle p_1, s_1\rangle = \langle p_2, s_2\rangle$. [1 mark]

(b) Prove [One-step determinacy for expressions]:
If $\langle e, s\rangle \longrightarrow \langle e_1, s_1\rangle$ and $\langle e, s\rangle \longrightarrow \langle e_2, s_2\rangle$ then $\langle e_1, s_1\rangle = \langle e_2, s_2\rangle$. [5 marks]

(c) Assume [Irrelevant store can be added]:
If $\langle e, s\rangle \longrightarrow \langle e_1, s_1\rangle$ and $\mathrm{dom}(s) \cap \mathrm{dom}(s_0) = \{\}$ then $\langle e, s \uplus s_0\rangle \longrightarrow \langle e_1, s_1 \uplus s_0\rangle$.

(d) Prove [Irrelevant store can be removed]:
If $\langle e, s \uplus s_0\rangle \longrightarrow \langle e_1, s_1\rangle$ and $\mathrm{loc}(e) \subseteq \mathrm{dom}(s)$ then there exists s' such that $\langle e, s\rangle \longrightarrow \langle e_1, s'\rangle$ and $s_1 = s' \uplus s_0$. [8 marks]

(e) Using (b), (c) and (d), prove [One-step confluence for store-disjoint threads]:
If $p = e_1 | e_2$, $\mathrm{loc}(e_1) \cap \mathrm{loc}(e_2) = \{\}$, $\langle p, s\rangle \twoheadrightarrow \langle p', s'\rangle$, and $\langle p, s\rangle \twoheadrightarrow \langle p'', s''\rangle$, then either $\langle p', s'\rangle = \langle p'', s''\rangle$ or $\exists\, p''', s'''.\ \langle p', s'\rangle \twoheadrightarrow \langle p''', s'''\rangle \wedge \langle p'', s''\rangle \twoheadrightarrow \langle p''', s'''\rangle$. [6 marks]