1997 Paper 5 Question 12

Semantics of Programming Languages

The phrases, P, of the language LC are specified by:

$$
\begin{aligned}
P &::= C \mid E \mid B \\
C &::= \mathbf{skip} \mid \ell := E \mid C\,;C \mid \mathbf{if}\ B\ \mathbf{then}\ C\ \mathbf{else}\ C \mid \mathbf{while}\ B\ \mathbf{do}\ C \\
E &::= n \mid !\ell \mid E\ \mathit{iop}\ E \\
B &::= \mathbf{true} \mid \mathbf{false} \mid E\ \mathit{bop}\ E
\end{aligned}
$$

where ℓ ranges over storage locations, n over integers, *iop* over integer-valued operations, and *bop* over boolean-valued operations. Describe the operational semantics of LC in terms of an inductively defined transition relation, $\to$, between configurations $\langle P, s\rangle$, where s is a finite partial function from locations to integers. State which are the *terminal* configurations and explain what it means for a configuration to be *stuck*. [6 marks]

Call a configuration $\langle P, s\rangle$ *sensible* if the set of locations on which s is defined, $dom(s)$, contains all the locations that occur in the phrase P. Prove by induction on the structure of P that for all s, if $\langle P, s\rangle$ is sensible then $\langle P, s\rangle$ is not stuck. [6 marks]

Prove by Rule Induction for $\to$ that if $\langle P, s\rangle \to \langle P', s'\rangle$ and $\langle P, s\rangle$ is sensible, then so is $\langle P', s'\rangle$ and $dom(s') = dom(s)$. Deduce that a stuck configuration can never be reached by a series of transitions from a sensible configuration. [8 marks]