1998 Paper 6 Question 12

Semantics of Programming Languages

What is meant by a *labelled transition system*?   [2 marks]

A language of commands, C, for interactive input/output is given by

$$C ::= \mathbf{skip} \mid \ell := \ell \mid \mathbf{getc}(\ell) \mid \mathbf{putc}(\ell) \mid C \ ; \ C$$
$$\mid \ \ \mathbf{if} \ \ell = \ell \ \mathbf{then} \ C \ \mathbf{else} \ C \mid \mathbf{while} \ \ell = \ell \ \mathbf{do} \ C$$

where ℓ ranges over some fixed set of locations for storing characters. The command $\ell_1 := \ell_2$ copies the contents of ℓ_2 to ℓ_1. The command $\mathbf{getc}(\ell)$ reads the next character from the standard input stream into ℓ. The command $\mathbf{putc}(\ell)$ writes the contents of ℓ to the standard output stream. The conditional and while-loop commands involve testing whether or not the contents of two locations are equal. The commands $\mathbf{skip}$ and $C_1 \ ; \ C_2$ have their usual meanings. Define an operational semantics for this language as a labelled transition system whose configurations are (command, state)-pairs, $\langle C, s \rangle$, and whose actions are of the form $get(c)$ (for reading a character c), $put(c)$ (for writing a character c), and τ (for transitions not involving input/output).   [9 marks]

Explain, without proof, in what sense this labelled transition system is deterministic.   [2 marks]

A finite list t of non-τ actions is a *trace* of $\langle C, s \rangle$ if there is a sequence of labelled transitions starting from $\langle C, s \rangle$ whose corresponding list of actions is equal to t once any τ-actions have been erased from it. Write $C \approx_{tr} C'$ to mean that for any state s, the configurations $\langle C, s \rangle$ and $\langle C', s \rangle$ have the same traces. Show that $C_1 \approx_{tr} C_2$, where

$$C_1 \ \stackrel{\text{def}}{=} \ \mathbf{getc}(\ell_1) \ ; \ \mathbf{putc}(\ell_1) \quad \text{and} \quad C_2 \ \stackrel{\text{def}}{=} \ \mathbf{getc}(\ell_1) \ ; \ \ell_2 := \ell_1 \ ; \ \mathbf{putc}(\ell_2).$$

[3 marks]

By considering $C_1 \ ; \ C$ and $C_2 \ ; \ C$ for a suitable C, or otherwise, show that $\approx_{tr}$ does not have the congruence property for this language.   [4 marks]