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

$$C ::= \text{skip} \mid \ell := \ell \mid \text{getc}(\ell) \mid \text{putc}(\ell) \mid C \; ; \; C$$

$$\mid \text{if } \ell = \ell \text{ then } C \text{ else } C \mid \text{while } \ell = \ell \text{ do } C$$

where $\ell$ ranges over some fixed set of locations for storing characters. The command $\ell_1 := \ell_2$ copies the contents of $\ell_2$ to $\ell_1$. The command $\text{getc}(\ell)$ reads the next character from the standard input stream into $\ell$. The command $\text{putc}(\ell)$ writes the contents of $\ell$ 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 $\text{skip}$ and $C_1 \; ; \; C_2$ have their usual meanings. Define an operational semantics for this language as a labelled transition system whose configurations are $(\text{command, state})$-pairs, $(C, s)$, and whose actions are of the form get($c$) (for reading a character $c$), put($c$) (for writing a character $c$), and $\tau$ (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-$\tau$ actions is a trace of $(C, s)$ if there is a sequence of labelled transitions starting from $(C, s)$ whose corresponding list of actions is equal to $t$ once any $\tau$-actions have been erased from it. Write $C \approx_{tr} C'$ to mean that for any state $s$, the configurations $(C, s)$ and $(C', s)$ have the same traces. Show that $C_1 \approx_{tr} C_2$, where

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

[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]