Semantics of Programming Languages

A language $L$ of expressions, $E$, for infinite lists of integers is given by:

$$E ::= \text{ints} \mid \text{from}(n) \mid n : E \mid \text{incr}(E)$$

where $n$ ranges over the integers. A binary relation of evaluation between $L$-expressions is inductively defined by the following axioms and rule:

$$(\downarrow_{\text{ints}}) \quad \text{ints} \Downarrow 0 : \text{incr}(\text{ints}).$$

$$(\downarrow_{\text{from}}) \quad \text{from}(n) \Downarrow n : \text{from}(n') \quad \text{where } n' = n + 1.$$  

$$(\downarrow:) \quad n : E \Downarrow n : E.$$  

$$(\downarrow_{\text{incr}}) \quad \frac{E \Downarrow n : E'}{\text{incr}(E) \Downarrow n' : \text{incr}(E')} \quad \text{where } n' = n + 1.$$  

Prove that for every $E$, there are unique $n$ and $E'$ such that $E \Downarrow n : E'$. [5 marks]

A binary relation $\mathcal{R}$ between $L$-expressions is called a bisimulation if whenever $(E_1, E_2) \in \mathcal{R}$ then $E_1 \Downarrow n : E'_1$ and $E_2 \Downarrow n : E'_2$ hold for some $n$ and some $(E'_1, E'_2) \in \mathcal{R}$. We write $E_1 \approx E_2$ if $E_1$ and $E_2$ are related by some bisimulation.

Prove that $\text{ints} \approx \text{from}(0)$. [5 marks]

Prove that $\approx$ has the congruence property for the language $L$, i.e. that if $E_1 \approx E_2$, then $E[E_1] \approx E[E_2]$ (where $E[E_1]$ is any $L$-expression containing an occurrence of $E_1$ and $E[E_2]$ denotes the result of replacing that occurrence by $E_2$). [Hint: show that $\approx$ is preserved by the operations $n : -$ and $\text{incr}(-)$, by constructing suitable bisimulations.] [5 marks]

State, with justification, whether or not $\approx$ is an equivalence relation. [5 marks]