9 Semantics of Programming Languages (PMS)

Consider the following language syntax:

Booleans $b \in \mathbb{B} = \{\textsf{true}, \textsf{false}\}$
Natural numbers $n \in \mathbb{N} = \{0, 1, ...\}$
Locations $\ell \in \mathbb{L} = \{l, l_0, l_1, l_2, ...\}$
Operations $op ::= + \mid - \mid \geq$
Expressions

$$e \quad ::= \quad n \mid b \mid e_1 \;\; op \;\; e_2 \mid \textsf{if} \;\; e_1 \;\; \textsf{then} \;\; e_2 \mid \ell := e \mid !\ell \mid \textsf{skip} \mid e_1; e_2 \mid \textsf{print} \;\; e$$

Types $T ::= \textsf{nat} \mid \textsf{bool} \mid \textsf{unit} \mid T\,\textsf{ref}$

(*a*) Define a reasonable operational semantics and type system for this syntax. Your operational semantics should be expressed as a relation

$$\langle e, s\rangle \xrightarrow{L} \langle e', s'\rangle$$

where s models the store and the label L is either n (for a **print** of that natural number) or τ (for an internal transition). Your type system should be expressed as a relation

$$\Gamma \vdash e : T$$

Make clear what s and Γ range over in your semantics.

[12 marks]

(*b*) Explain the main design choices you made in part (*a*), giving alternative rules or examples (of expressions, transitions, or derivations of transitions or typing) as appropriate. Discuss whether your semantics has type preservation and progress properties. [8 marks]