10 Semantics of Programming Languages (PMS)

Consider the following syntax:

Booleans $b \in \mathbb{B} = \{\textsf{\textbf{true}}, \textsf{\textbf{false}}\}$
Integers $n \in \mathbb{Z} = \{..., -1, 0, 1, ...\}$
Variables $x \in \mathbb{X} = \{x, y, ...\}$
Expressions $e ::= b \mid n \mid x \mid \textsf{\textbf{fn}}\ x \to e \mid e_1\ e_2 \mid \textsf{\textbf{print}}\ e \mid \textsf{\textbf{skip}}$
(considered up to alpha equivalence, with x binding in e in $\textsf{\textbf{fn}}\ x \to e$)

The set of free variables of an expression $\mathrm{fv}(e)$ are defined in the normal way as follows.

$$\begin{array}{lcl}
\mathrm{fv}(b) & = & \{\} \\
\mathrm{fv}(n) & = & \{\} \\
\mathrm{fv}(x) & = & \{x\} \\
\mathrm{fv}(\textsf{\textbf{fn}}\ y \to e) & = & \mathrm{fv}(e) - \{y\} \\
\mathrm{fv}(e_1\ e_2) & = & \mathrm{fv}(e_1) \cup \mathrm{fv}(e_2) \\
\mathrm{fv}(\textsf{\textbf{print}}\ e) & = & \mathrm{fv}(e) \\
\mathrm{fv}(\textsf{\textbf{skip}}) & = & \{\}
\end{array}$$

(*a*) Define capture-avoiding substitution $\{e/x\}e'$. [3 marks]

(*b*) Define a small-step right-to-left call-by-value operational semantics for this syntax. Your semantics should be expressed as a relation

$$e \xrightarrow{L} e'$$

where the label L is either n (for a **print** of that integer) or τ (for an internal transition). [5 marks]

(*c*) Explain how a call-by-name semantics would differ, giving any changes required to the rules and giving an example expression that has different output in the two semantics (you should give its transitions in each but need not give their derivations). [3 marks]

(*d*) We are normally interested in closed programs (with no free variables). Prove, with respect to your call-by-value semantics of part (*b*), that if e is closed and $e \xrightarrow{L} e'$ then e' is closed. You can omit the cases for **print**. [9 marks]