2009 Paper 6 Question 9

Semantics of Programming Languages

Consider the following syntax for a pure untyped functional language.

Booleans $b \in \mathbb{B} = \{\textsf{true}, \textsf{false}\}$
Integers $n \in \mathbb{Z} = \{..., -1, 0, 1, ...\}$
Variables $x \in \mathbb{X}$ for a set $\mathbb{X} = \{\mathrm{x}, \mathrm{y}, \mathrm{z}, ...\}$
Operations $op ::= + \mid \geq$
Expressions

$$e ::= \textsf{skip} \mid n \mid b \mid e_1 \;\; op \;\; e_2 \mid \textsf{if} \;\; e_1 \;\; \textsf{then} \;\; e_2 \;\; \textsf{else} \;\; e_3 \mid \textsf{fn} \;\; x \Rightarrow e \mid e_1 \; e_2 \mid x \mid \textsf{fix} \; e$$

The language supports recursion with a fixed-point operator $\textsf{fix} \; e$, which has semantics defined by the rule below.

$$\frac{}{\textsf{fix}\; e \longrightarrow e(\textsf{fix}\; e)}$$

(*a*) Give the semantic rules for function application for call-by-value, call-by-name, and full-beta reduction for this language (do not give the rules for binary operators, conditional, or fix). You should define a small-step reduction relation $e \longrightarrow e'$, stating precisely what notion of values v you are using. [10 marks]

(*b*) For the call-by-value semantics, characterise the expressions e from the grammar above that have an *immediate* runtime error in their outermost (top-level) construct. [3 marks]

(*c*) For each pair of semantics (call-by-value and call-by-name, call-by-name and full-beta, and full-beta and call-by-value), give an expression with different possible termination behaviours in each element of the pair. [4 marks]

(*d*) For each of your three semantics, explain a disadvantage in using that semantics for a programming language. [3 marks]