10 Semantics of Programming Languages (SS)

This question is about a variation on a fragment of the L2 language in which functions take two arguments. The language has the following expressions:

\[ e ::= x \mid \text{fn}(x_1, x_2) \Rightarrow e \mid e_0(x_1, e_2) \mid n \]

where \( x \) ranges over variables and \( n \) ranges over integers. As usual, \( \text{fn}(x, y) \Rightarrow e \) is binding: we work up-to \( \alpha \)-equivalence and require that \( x \) and \( y \) are different.

(a) Write down a call-by-name operational semantics for this language. [2 marks]

(b) Consider the following type system. The types are

\[ T ::= \text{int} \mid \text{ret} \mid (T_1, T_2) \rightarrow \text{ret} \]

A context \( \Gamma \) is a finite partial function from variables to types. The type system is given by the following rules:

\[
\begin{align*}
\frac{}{\Gamma, x : T, \Gamma' \vdash x : T} & \quad \frac{\Gamma \vdash e_0 : (T_1, T_2) \rightarrow \text{ret} \quad \Gamma \vdash e_1 : T_1 \quad \Gamma \vdash e_2 : T_2}{\Gamma \vdash e_0(x_1, e_2) : \text{ret}} \\
\frac{}{\Gamma \vdash n : \text{int}} & \quad \frac{\Gamma, x_1 : T_1, x_2 : T_2 \vdash e : \text{ret}}{\Gamma \vdash \text{fn}(x_1, x_2) \Rightarrow e : (T_1, T_2) \rightarrow \text{ret}}
\end{align*}
\]

(The idea is that \((T_1, T_2) \rightarrow \text{ret}\) is a type of functions taking arguments of type \(T_1\) and \(T_2\). However, there are no expressions of type \(\text{ret}\) in the empty context, and so rather than returning a result you pass it to a ‘continuation’.)

(i) Find a type \(T\) for which \(\vdash \text{fn}(x, k) \Rightarrow k(3, x) : T\), giving a derivation. [3 marks]

(ii) Give a derivation of the following judgement: [2 marks]

\[ k : (\text{int, ret}) \rightarrow \text{ret} \vdash \text{fn}(x, l) \Rightarrow l(7, k) : (\text{int, (int, (int, ret) \rightarrow ret) \rightarrow ret}) \rightarrow \text{ret} \rightarrow \text{ret} \]

(c) Prove the following ‘progress’ theorem for this language: [6 marks]

If \(\vdash e : T\) then either \(e = (\text{fn}(x, y) \Rightarrow e')\), or \(e\) is an integer, or there is \(e'\) such that \(e \longrightarrow e'\).

(d) We now consider the situation where there is a type \(\text{posint}\) of positive integers which is a subtype of \(\text{int}\).

Define a subtyping relation \(<:\) and extend the type system to accommodate it. Demonstrate it by giving a derivation of the following judgement:

\[ k : (\text{int, ret}) \rightarrow \text{ret} \vdash \text{fn}(x, l) \Rightarrow l(7, k) : (\text{int, (int, (posint, ret) \rightarrow ret) \rightarrow ret}) \rightarrow \text{ret} \rightarrow \text{ret} \]