1996 Paper 7 Question 10

Types

Explain the term *minimal type* and discuss its importance in typechecking algorithms for type systems with subtyping. What is the difference between a minimal type and a principal typing? [6 marks]

Write subtyping and typing algorithms (*either* as syntax-directed systems of inference rules *or* as pseudo-code) for the following "core" of the simply typed lambda-calculus with subtyping.

$$\begin{array}{rcl} e & ::= & x \\ & & \mathit{fun}(x \in T)e \\ & & e_1\ e_2 \end{array}$$

$$\begin{array}{rcl} T & ::= & T_1 \to T_2 \\ & & \mathit{Top} \end{array}$$

Your algorithms need not handle records or booleans. [6 marks]

Suppose that we add to this calculus a type $\mathit{Box}(T)$ for each type T, and the expression constructors

$$\begin{array}{rcl} e & ::= & \ldots \\ & & \mathit{box}\ e \\ & & \mathit{contents}\ e \\ & & e_1 \leftarrow e_2 \end{array}$$

with the following evaluation rules:

$$\frac{e \Downarrow r}{\mathit{box}\ e \Downarrow \mathit{box}\ r}$$

$$\frac{e \Downarrow \mathit{box}\ r}{\mathit{contents}\ e \Downarrow r}$$

$$\frac{e_1 \Downarrow \mathit{box}\ r_1 \qquad e_2 \Downarrow r_2}{e_1 \leftarrow e_2 \Downarrow \mathit{box}\ r_2}$$

Write sound typing and subtyping rules for these constructs. [5 marks]

Now suppose that we add to this calculus the type variables and bounded universal quantification of System $F_{\leqslant}$. Indicate how your typing and/or subtyping rules must change (while remaining sound!) so that the expression

$$\mathit{fun}(X \leqslant \mathit{Box}(\mathit{Top} \to \mathit{Top}))\ \mathit{fun}(x \in X)\ x \leftarrow (\mathit{fun}(y \in \mathit{Top})\ y)$$

has the type

$$\mathit{All}(X \leqslant \mathit{Box}(\mathit{Top} \to \mathit{Top}))\ X \to X$$

[3 marks]