2001 Paper 7 Question 13

Types

(*a*) What does it mean to say that an ML type scheme σ *generalises* an ML type τ? Writing $\sigma \succ \tau$ for this relation and defining

$$\sigma_1 = \forall\{\alpha, \beta\}(\alpha \to \beta) \qquad \sigma_2 = \forall\{\alpha\}(\alpha \to \beta)$$
$$\tau_1 = (\alpha \to \beta) \to \alpha \qquad \tau_2 = (\beta \to \alpha) \to \beta$$

say whether or not $\sigma_i \succ \tau_j$ holds for each of the four possibilities. [5 marks]

(*b*) Give the axioms and rules for inductively generating ML typing judgements of the form

$$\Gamma \vdash M : \tau$$

where $\Gamma = (\Gamma_{\mathrm{tv}}, \Gamma_{\mathrm{ta}})$ with Γ_{tv} a finite set of type variables, Γ_{ta} a finite function mapping some variables to type schemes whose free type variables are in Γ_{tv}, and τ is a type whose type variables are in Γ_{tv}. You may restrict attention to expressions M involving variables, boolean values, conditionals, function abstraction and application, and let-expressions. [8 marks]

(*c*) Explain why the expressions $\mathbf{let}\, x = M \,\mathbf{in}\, M'$ and $(\lambda\, x\, (M'))M$ can have different typing properties in the ML type system even though their evaluation behaviour is the same; illustrate your answer by taking M to be $\lambda\, y\, (y)$ and M' to be $x\, x$. [7 marks]