Types

Let x range over a set of identifiers and α range over a set of type variables. Now suppose we have a set of types, τ, and a set of type schemes, σ, given by

$$\begin{aligned} \tau &::= \alpha \quad | \quad \tau \to \tau \quad | \quad \tau \; list \\ \sigma &::= \forall \alpha_1, \ldots, \alpha_n \, (\tau) \end{aligned}$$

and a language of terms, M, given by

$$\begin{aligned} M \;\; ::= \;\; & x \;\; | \;\; \lambda x \, (M) \;\; | \;\; M \; M \;\; | \;\; \texttt{let} \; x = M \; \texttt{in} \; M \;\; | \;\; \texttt{nil} \;\; | \;\; M :: M \\ | \;\; & \texttt{case} \; M \; \texttt{of} \; \texttt{nil} \Rightarrow M \mid x :: x \Rightarrow M \end{aligned}$$

(*a*) Define the relation of specialisation, $\tau \prec \sigma$, between types τ and type schemes σ. [3 marks]

(*b*) Give the ML-like type inference rules for judgements of the form $\Gamma \vdash M : \tau$, and explain why λ-bound variables cannot be used polymorphically within a function abstraction, while `let`-bound variables can within a local declaration.

Hint: Consider the terms $\lambda f (f \; f)$ and $\texttt{let} \; f = \lambda x (x) \; \texttt{in} \; f \; f$.

[8 marks]

(*c*) Briefly explain what is meant by *capture-avoiding substitution* for type schemes. [2 marks]

(*d*) Prove that for all τ, all σ and all substitutions for type schemes S, if $\tau \prec \sigma$ holds, then also $S(\tau) \prec S(\sigma)$.

Hint: Use the following property of simultaneous substitution:

$$(\tau[\tau_1/\alpha_1, \ldots, \tau_n/\alpha_n])[\vec{\tau'}/\vec{\alpha'}] = \tau[\vec{\tau'}/\vec{\alpha'}][\tau_1[\vec{\tau'}/\vec{\alpha'}]/\alpha_1, \ldots, \tau_n[\vec{\tau'}/\vec{\alpha'}]/\alpha_n]$$

which holds, provided that for each i, α_i is distinct from the type variables $\vec{\alpha'}$, and α_i does not occur in type schemes $\vec{\tau'}$.

[7 marks]