

1993 Paper 8 Question 11

Types

Explain what is meant by the statement ‘every closed, typable expression in ML possesses a *principal* type’. Does a similar property hold for the second order lambda calculus λ^2 ? [4 marks]

What is meant by a *type scheme* in ML? Give the rules for inductively defining ML type assertions of the form

$$\Gamma \vdash M : \rho$$

where ρ is a type scheme, Γ is a finite function from identifiers to type schemes, and M is an ML expression built up from identifiers using let-expressions, lambda abstractions and function applications. [9 marks]

This fragment of ML is augmented by fixed-point expressions $\text{fix } x.M$, which are typed according to the rule

$$\frac{\Gamma, x : \rho \vdash M : \rho}{\Gamma \vdash \text{fix } x.M : \rho} \quad (\text{FIX})$$

(Free occurrences of x in M become bound in $\text{fix } x.M$.) Show that the closed expression $\text{fix } x.\lambda y.(xx)y$ is typable in this augmented system. Hint: find a type scheme ρ for which $x : \rho \vdash \lambda y.(xx)y : \alpha \rightarrow \beta$ holds. [4 marks]

Is the expression typable if the use of the rule (FIX) is restricted by requiring ρ to be a type rather than a type scheme? [3 marks]