

## 1994 Paper 8 Question 13

### Types

Explain what is meant by saying that a programming language is

- (a) strongly typed
- (b) monomorphic
- (c) polymorphic

[4 marks]

Briefly describe some ways in which polymorphism may arise in programming languages. [5 marks]

Consider extending the ML type inference system with a new type constant  $\omega$  and a type inference rule

$$\text{(UNIV)} \frac{}{\Gamma \vdash M : \omega}$$

where  $M$  is any expression and  $\Gamma$  is any context assigning types to a finite set of identifiers that includes the free identifiers in  $M$ . Prove that

$$\vdash FF : (\omega \rightarrow \sigma) \rightarrow \sigma$$

holds in this extended system, where  $\sigma$  is any type and  $F$  is the expression

$$\lambda y. \lambda x. x((yy)x) \quad [6 \text{ marks}]$$

Do closed expressions possess *principal* types in this extension of ML?

[Hint: consider the possible types  $\lambda x. x$  may possess in this system.] [5 marks]