

2004 Paper 5 Question 10

Foundations of Functional Programming

Pure lambda-calculus does not have any built-in control structures or arithmetic. It just has lambda-expressions.

- (a) Using an untyped pure lambda calculus, explain how to model the natural numbers $0, 1, \dots$ together with a test for zero and addition and multiplication operations. [7 marks]
- (b) How can recursive function definitions be expressed in pure lambda calculus? Give a lambda-term for any special things that you need to introduce. [4 marks]
- (c) Suppose you are provided with a lambda expression that works with numbers in the sense of part (a) above and which will find the predecessor of any non-zero number. Express the factorial function in terms of the primitives that you introduced in parts (a) and (b). [5 marks]
- (d) Explain how a typical polymorphic typing system would respond to the various lambda-expressions you have given. If there were any that would fail to type-check, explain why: if everything you have written could be given valid polymorphic types, show what those types would be for a couple of the key expressions. [4 marks]

Credit will be given for clarity and conciseness of presentation, and justification of why your lambda expressions apply in the general case will be preferred to sets of examples that merely illustrate them in particular cases.