UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part III
BSc Honours Degree in Mathematics and Computer Science Part III
MSci Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C382

TYPE SYSTEMS FOR PROGRAMMING LANGUAGES

Monday 14 May 2001, 14:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1) a. Give the definition of

   i. Lambda Terms.

   ii. Curry types.

   iii. Curry type assignment for the Lambda Calculus.

   b. Give, in functional programming language notation, the algorithm that calculates the principal Curry pair for lambda terms.

   c. Give a definition of substitution on types, and show that substitution is sound with respect to type assignment in Curry's system.

   d. What is *Polymorphism*? Give an extension of the syntax definition for Lambda Calculus, as done for ML, that allows you to introduce polymorphism. Give the natural extension to Curry's type assignment system for the Lambda Calculus that deals with this new construct.

   e. What is *Recursion*? Give an extension of the syntax definition for Lambda Calculus, as done for ML, that allows you to introduce polymorphism. Give the natural extension to Curry's type assignment system for the Lambda Calculus that deals with this new construct.

The five parts carry, respectively, 10%, 25%, 35%, 15%, and 15% of the marks.

2) a. Show that $(\lambda yg.g(yy))(\lambda yg.g(yy))$ is a fixed point constructor for the Lambda Calculus.

b. Explain why, although it is in principle possible to translate (compile) all functional programs into lambda terms, such a translation would not preserve assignable types. How is this problem overcome?

c. For the following terms, answer the question whether they are typeable in, respectively, Curry's type assignment system for the Lambda Calculus, in the Polymorphic type assignment system for the Lambda Calculus, and in the Intersection type assignment system for the Lambda Calculus (with a strict type).

For each, if not, motivate your answer, and if yes, give a derivation.

i. $\lambda x.xx$?

ii. $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$?

d. Is type assignment decidable in these three systems? For Curry's system and the Intersection system, motivate your answer.

The three parts carry 20%, 20%, 45%, and 15% of the marks, respectively.

3) a. Give the definition of

i. Terms.

ii. Rewrite rules and reduction.

iii. Curry type assignment for the Term Rewriting Systems ($\vdash_{\mathcal{E}}$).

b. What is, essentially, the difference between Milner's and Mycroft's type assignment system for ML? Give the definition of type assignment to rewrite rules, as presented in the course. On which approach is this based: Milner's of Mycroft's?

Specify what difference using the other approach would bring.

c. Using the properties

i. If $pp_{\mathcal{E}} M = \langle P, \pi \rangle$, and for the replacement R there are B and σ such that $B \vdash_{\mathcal{E}} t^{\mathrm{R}} : \sigma$, then there is a substitution S, such that $S\pi = \sigma$, and for every statement $x : \rho \in P$: $B \vdash_{\mathcal{E}} x^{\mathrm{R}} : S\rho$.

ii. If $B \vdash_{\mathcal{E}} t : \sigma$, and R is a replacement and B' a basis such that for every statement $x : \rho \in B$: $B' \vdash_{\mathcal{E}} x^{\mathrm{R}} : \rho$, then $B' \vdash_{\mathcal{E}} t^{\mathrm{R}} : \sigma$.

iii. Soundness of Substitution.

iv. The principal pair property.

show the Subject Reduction Theorem.

The three parts carry, respectively, 30%, 30%, and 40% of the marks.

4) a. Give the definition of

   i. Intersection types.

   ii. Intersection type assignment for the Lambda Calculus.

   b. Derive the principal (intersection) pair for $\lambda xyz.xz(yz)$. Use this to derive a type for S that this is not derivable in Curry's system.

   c. Derive an intersection type for $(\lambda xyz.xz(yz))(\lambda xy.xz)$ that is not derivable in Curry's system. (Your answer to the previous part need not be part of the derivation you are looking for here.)

   d. Give the characterisation using intersection types of

      i. normalisation

      ii. head-normalisation

      iii. strong normalisation

   e. Show that type assignment is closed for subject expansion in the intersection type assignment system; you can use an intuitive argument rather than a formal proof.

The five parts carry each 20% of the marks.