UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2002

BEng Honours Degree in Computing Part III
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

PAPER C382

TYPE SYSTEMS FOR PROGRAMMING LANGUAGES

Tuesday 7 May 2002, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions
Calculators not required

1 i. Give the definition of the mapping $\langle\rangle_\lambda$, that maps terms in a Combinatory System to Lambda terms, and of the mapping $[\![\,]\!]_{\mathrm{CL}}$, that maps Lambda terms to terms in Combinatory Logic.

ii. Show that $[\![\lambda xy.xy]\!]_{\mathrm{CL}} = \mathsf{S}(\mathsf{S}(\mathsf{KS})(\mathsf{S}(\mathsf{KK})\mathsf{I}))(\mathsf{KI})$, and show that $\langle[\![\lambda xy.xy]\!]_{\mathrm{CL}}\rangle_\lambda \twoheadrightarrow_\beta \lambda xy.xy$ by reducing the first term (you do not need to show each individual step, but could do as many as you like 'in parallel').

iii. Show that $\emptyset \vdash_{\mathcal{E}_{\mathrm{CL}}} \mathsf{S}(\mathsf{S}(\mathsf{KS})(\mathsf{S}(\mathsf{KK})\mathsf{I}))(\mathsf{KI}) : (\sigma\to\tau)\to\sigma\to\tau$.

The three parts carry, respectively, 20%, 60%, and 20% of the marks.

2 i. Give the syntax definition for ML-terms, and the essential rules in the notion of reduction of the ML-language.

ii. Give the syntax definition for ML-types, and give the type assignment rules for the Milner type assignment system.

iii. What is, essentially, the difference between Milner's and Mycroft's type assignment system for ML? Give the type assignment rules that are different, and explain how they differ. Specify how this difference reflects on the principal type algorithm.

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.

iv. Consider the algebraic data type *Int*

$$n ::= 0 \mid (\mathit{Succ}\, n).$$

and the program (in functional style), with addition and multiplication:

$$\begin{array}{lll} \mathit{Add}\ 0 & y = & y, \\ \mathit{Add}\ (\mathit{Succ}\ x) & y = & \mathit{Succ}\ (\mathit{Add}\ x\ y) \\ \mathit{Mul}\ 0 & y = 0, & \\ \mathit{Mul}\ (\mathit{Succ}\ x) & y = \mathit{Add}\, y\ (\mathit{Mul}\ x\ y) & \\ \mathit{Mul}\, 5\ 3 & & \end{array}$$

Assuming 0 and *Succ* are represented by terms with the same name (appearance), translate these definitions into an ML-term.

The four parts carry each, respectively, 25%, 30%, %, 25%, and 20% of the marks.

3 i. Give, for the Term Rewriting Systems, the definition of

A Terms.

B Rewrite rules and reduction.

C Curry type assignment ($\vdash_{\mathcal{E}}$).

ii. Given the term rewriting system

$$
\begin{array}{lcl}
\mathsf{I}\,x & \to & x \\
\mathsf{K}\,x\,y & \to & x \\
\mathsf{Z}\,x\,y & \to & y \\
\mathsf{B}\,x\,y\,z & \to & x\,(y\,z) \\
\mathsf{C}\,x\,y\,z & \to & x\,z\,y \\
\mathsf{S}\,x\,y\,z & \to & x\,z\,(y\,z)
\end{array}
$$

Give an environment that makes these rules typeable.

iii. Add the following rules to the system above.

$$
\begin{array}{lcl}
\mathsf{S}\,(\mathsf{K}\,x)\,(\mathsf{K}\,y) & \to & \mathsf{K}\,(x\,y) \\
\mathsf{S}\,(\mathsf{K}\,x)\,y & \to & \mathsf{B}\,x\,y \\
\mathsf{S}\,x\,(\mathsf{K}\,y) & \to & \mathsf{C}\,x\,y
\end{array}
$$

Show that the system is still typeable using the same environment.

iv. Add now also the rule

$$\mathsf{S}\,(\mathsf{K}\,x)\,\mathsf{I} \to x$$

Show that the system is no longer typeable using the same environment. What would you have to change to make the system typeable?

The four parts carry, respectively, 30%, 20%, 30%, and 20% of the marks.

4 i. Give the definition of

  A Intersection types.

  B Intersection type assignment for the Lambda Calculus.

ii. Give the characterisation using intersection types of

  A normalisation

  B head-normalisation

  C strong normalisation

iii. Does $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$ have a single principal intersection type? If yes, give a derivation that derives that type; if no, explain why.
Show that $(\omega{\to}\varphi){\to}\varphi$ is a type for $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$.

iv. Does $\lambda x.xx$ have a single principal intersection type? If yes, give a derivation that derives that type; if no, explain why.
Show that $(\omega{\to}\varphi){\to}\varphi$ is a type for $\lambda x.xx$.

The four parts carry, respectively, 35%, 15%, 30%, and 20% of the marks.

End of Paper