# 1997 Paper 6 Question 9

## Foundations of Functional Programming

The $\lambda_I$-calculus is a variant of the $\lambda$-calculus. The *terms* of the $\lambda_I$-calculus, known as $\lambda_I$-*terms*, are constructed recursively from a given set $V$ of variables; the $\lambda_I$-terms take one of the following forms:

| | |
|---|---|
| $x$ | variable |
| $\lambda x.M$ | abstraction, where $M$ is a $\lambda_I$-term and $x \in \mathrm{FV}(M)$ |
| $MN$ | application, where $M$ and $N$ are $\lambda_I$-terms. |

The set of free variables $\mathrm{FV}(M)$, $\beta\eta$-equality and $\beta\eta$-reduction are defined in a similar fashion to the corresponding $\lambda$-calculus definitions.

($a$) Define an equality-preserving translation from $\lambda_I$-terms to combinators constructed using **I**, **B**, **C** and **S**. Indicate why these combinators are not enough to express the usual $\lambda$-calculus. [6 marks]

($b$) Demonstrate the translation using the $\lambda_I$-term $\lambda x.\lambda y.(xMMy)$, where $M$ is the identity function $\lambda z.z$. [3 marks]

($c$) Define the Church numerals for the usual $\lambda$-calculus, and identify those numerals which are not $\lambda_I$-terms. [3 marks]

($d$) By adapting the definition of the Church numerals, define $\lambda_I$-terms $\overline{n}$ for each $n \geqslant 0$ such that

$$\overline{n}\,MN = M^n(N) \text{ for } n \geqslant 1 \text{ and arbitrary } \lambda_I\text{-terms } M, N$$

$$\overline{0}\,\overline{m}\,\overline{n} = \overline{n} \text{ for arbitrary } m, n \geqslant 0.$$

Show that these equalities are indeed satisfied. [Hint: the $\lambda_I$-term shown in ($b$) may help to resolve a particular case.] [8 marks]