2008 Paper 5 Question 9

Foundations of Functional Programming

(*a*) Define the translation of the *call-by-name* λ-calculus into continuation passing style. [9 marks]

(*b*) How does the translation differ for the *call-by-value* λ-calculus? [2 marks]

(*c*) Now consider extending the *call-by-name* λ-calculus with exceptions:

$$\begin{array}{rcl} M & ::== & \mathsf{try}\ M\ \mathsf{catch}\ M \\ & | & \mathsf{raise} \\ & | & \lambda x.\, M \\ & | & M\, M \\ & | & x \end{array}$$

where it reduces in the following way:

$$\begin{array}{rcl} \mathsf{try\ raise\ catch}\ M & \rightarrow & M \\ \mathsf{try}\ \lambda x.M_1\ \mathsf{catch}\ M_2 & \rightarrow & \lambda x.M_1 \\ \mathsf{raise}\ M & \rightarrow & \mathsf{raise} \end{array}$$

Show how to translate this language into pure λ-calculus using continuations.

[Hint: Use two continuations: one for the exceptional case, and one for the normal case.]

[9 marks]