1995 Paper 9 Question 12

Types

Consider the following datatype and function declarations in Standard ML:

```
datatype tree = Leaf | Node of tree * tree ;
fun iter x f Leaf = x
  | iter x f (Node(y,z)) = f(iter x f y)(iter x f z) ;
```

You are required to encode the datatype `tree` as a closed type τ in the second-order lambda calculus, $\lambda 2$. Find a suitable type τ and closed $\lambda 2$ terms in β-normal form, L, N, and I say, corresponding to `Leaf`, `Node` and `iter` respectively. You should demonstrate for your choices that

$$
\begin{aligned}
&\vdash L : \tau \\
&\vdash N : \tau \to \tau \to \tau \\
&\vdash I : \forall \alpha . \alpha \to (\alpha \to \alpha \to \alpha) \to \tau \to \alpha
\end{aligned}
$$

are derivable typing assertions, and that $I_\alpha x f L$ and $I_\alpha x f (N y z)$ are β-convertible to the $\lambda 2$ terms corresponding respectively to the right-hand sides of the clauses in the declaration of `iter`. [14 marks]

Now add to the above Standard ML declarations the function declarations

```
fun rev Leaf = Leaf
  | rev (Node(y,z)) = Node(rev z, rev y) ;
fun div Leaf = Leaf
  | div (Node(y,z)) = div(Node(z,y)) ;
```

Using I, or otherwise, show that there is a closed $\lambda 2$ term of type $\tau \to \tau$, R say, for which RL and $R(Nyz)$ are β-convertible to the $\lambda 2$ terms corresponding respectively to the right-hand sides of the clauses in the declaration of `rev`. Is there a closed $\lambda 2$ term D with similar properties for the declaration of `div`? [6 marks]