1999 Paper 9 Question 13

Types

A common idiom in typed programming languages is an *option type*, adding an additional value to an existing type. For example, in ML one might use

```
datatype 'a Option = None
                   | Some of 'a
```

Give a PLC encoding of `Option`, i.e. a PLC type Opt_α with free type variable α and suitable PLC terms such that

$$\{\alpha\}, \emptyset \vdash \mathrm{None}_\alpha : \mathrm{Opt}_\alpha$$
$$\{\alpha\}, \emptyset \vdash \mathrm{Some}_\alpha : \alpha \to \mathrm{Opt}_\alpha.$$

Give a typing derivation for None_α. [8 marks]

Any function, say $f : \gamma \to \delta$, can be lifted to a function of type $\mathrm{Opt}_\gamma \to \mathrm{Opt}_\delta$ that takes None_γ to None_δ and is as f elsewhere. Give a suitable PLC term $\mathrm{Lift}_{\gamma\delta}$ such that

$$\{\gamma, \delta\}, \emptyset \vdash \mathrm{Lift}_{\gamma\delta} : (\gamma \to \delta) \to (\mathrm{Opt}_\gamma \to \mathrm{Opt}_\delta).$$

Show the beta-equivalence

$$(\mathrm{Lift}_{\gamma\delta}\, f)\,(\mathrm{Some}_\gamma\, x) \;=_\beta\; \mathrm{Some}_\delta\,(f\,x)$$

[7 marks]

Similarly, functions $f : \gamma \to \mathrm{Opt}_\delta$ and $g : \delta \to \mathrm{Opt}_\epsilon$ can be composed. Give a suitable PLC term such that

$$\{\gamma, \delta, \epsilon\}, \emptyset \vdash \mathrm{Compose}_{\gamma\delta\epsilon} : (\gamma \to \mathrm{Opt}_\delta) \to (\delta \to \mathrm{Opt}_\epsilon) \to (\gamma \to \mathrm{Opt}_\epsilon).$$

[5 marks]