

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

$$\begin{aligned} \{\alpha\}, \emptyset &\vdash \text{None}_\alpha : \text{Opt}_\alpha \\ \{\alpha\}, \emptyset &\vdash \text{Some}_\alpha : \alpha \rightarrow \text{Opt}_\alpha. \end{aligned}$$

Give a typing derivation for None_α . [8 marks]

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

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

Show the beta-equivalence

$$(\text{Lift}_{\gamma\delta} f) (\text{Some}_\gamma x) =_\beta \text{Some}_\delta (f x)$$

[7 marks]

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

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

[5 marks]