Types and Programming Languages, Exercise 3

Philip Wadler


TAPL refers to *Types and Programming Languages* by Benjamin Pierce.

1. Update your answer to Exercise 2.1 [a–c] (‘Using your implementation . . . write and test the following’) to simply-typed lambda calculus. Since you have not yet implemented simply-typed lambda calculus, use a typed functional language such as O’Caml or Haskell to test and debug your solutions.

Hint: The type of the Church numerals may vary depending on how they are used. For example, depending on the definitions you use, addition may have type $C \rightarrow C \rightarrow C$, while multiplication may have type $C_2 \rightarrow C_2 \rightarrow C_2$, where $C = (N \rightarrow N) \rightarrow N \rightarrow N$, and $C_2 = (C \rightarrow C) \rightarrow C \rightarrow C$, and $N$ is the type of the naturals.