Programming Languages and Types

Exercise 12

Yi Dai

January 29, 2013

1 Simply-Typed λ-Calculus

1.1 Typing Derivation

1.2 Programming with Extensions

1. Define the subtraction function *sub* in the simply-typed λ-calculus extended with Peano numbers (**0** and **succ**) and fixed point operator **fix**.[1]

$$
\begin{array}{l}
\textbf{fix}\ (\lambda\ fsub : (\mathbf{N} \to \mathbf{N} \to \mathbf{N}) \to (\mathbf{N} \to \mathbf{N} \to \mathbf{N})\ . \\
\qquad\quad \lambda\ m : \mathbf{N}\ .\ \lambda\ n : \mathbf{N} \\
\qquad\qquad \textbf{if}\ (\textbf{iszero}\ m) \\
\qquad\qquad\quad \textbf{then}\ \mathbf{0} \\
\qquad\qquad\quad \textbf{else}\ (\textbf{succ}\ (fsub\ (\textbf{pred}\ m)\ n))\ )
\end{array}
$$

2 System-$\mathcal{F}$

2.1 Parametric Polymorphism

2.2 Typing Church-Encodings

[1] I apologize for giving the wrong type ($\mathbf{N} \to \mathbf{N} \to \mathbf{N}$) to the parameter $fsub$ during the exercise session.