Simply Typed Lambda Calculus with Parametric Polymorphism

CS 2520R HW 1

1 Syntax

$$
\begin{aligned}
e ::=\ & x && \text{Variable} \\
\mid\ & \lambda x : \tau.\ e && \text{Abstraction} \\
\mid\ & e_1\ e_2 && \text{Application} \\
\mid\ & \Lambda X.\ e && \text{Type abstraction} \\
\mid\ & e\ [\tau] && \text{Type application}
\end{aligned}
$$

$$
\begin{aligned}
\tau ::=\ & X && \text{Type variable} \\
\mid\ & \tau_1 \to \tau_2 && \text{Function type} \\
\mid\ & \forall X.\ \tau && \text{Universal type}
\end{aligned}
$$

2 Typing Rules

$$
\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau}\ \text{T-Var}
\qquad
\frac{\Gamma, x : \tau_1 \vdash e : \tau_2}{\Gamma \vdash \lambda x : \tau_1.\ e : \tau_1 \to \tau_2}\ \text{T-Abs}
\qquad
\frac{\Gamma \vdash e_1 : \tau_1 \to \tau_2 \qquad \Gamma \vdash e_2 : \tau_1}{\Gamma \vdash e_1\ e_2 : \tau_2}\ \text{T-App}
$$

$$
\frac{\Gamma, X \vdash e : \tau}{\Gamma \vdash \Lambda X.\ e : \forall X.\ \tau}\ \text{T-TAbs}
\qquad
\frac{\Gamma \vdash e : \forall X.\ \tau_1}{\Gamma \vdash e\ [\tau_2] : \tau_1[X := \tau_2]}\ \text{T-TApp}
$$

3 Operational Semantics

We define a small-step operational semantics with the following reduction rules:

$$
\frac{}{(\lambda x : \tau.\ e_1)\ e_2 \longrightarrow e_1[x := e_2]}\ \text{E-App}
\qquad
\frac{}{(\Lambda X.\ e)\ [\tau] \longrightarrow e[X := \tau]}\ \text{E-TApp}
\qquad
\frac{e_1 \longrightarrow e_1'}{e_1\ e_2 \longrightarrow e_1'\ e_2}\ \text{E-App1}
$$

$$
\frac{e_2 \longrightarrow e_2'}{v_1\ e_2 \longrightarrow v_1\ e_2'}\ \text{E-App2}
\qquad
\frac{e \longrightarrow e'}{e\ [\tau] \longrightarrow e'\ [\tau]}\ \text{E-TApp1}
$$

Where v_1 represents a value (either a lambda abstraction or a type abstraction).

References:

[1] https://www3.nd.edu/ dchiang/teaching/pl/2022/f.html
[2] https://www.cs.utexas.edu/ bornholt/courses/cs345h-24sp/lectures/8-system-f/