1 Extended Grammar

Here are some additional terms not defined in the core grammar.

$$
\begin{aligned}
e ::=\ & f = \lambda x : \tau . e \\
\mid\ & f x \\
\mid\ & \texttt{val}\ x : \tau = e \\
\mid\ & \texttt{let}\ var = e\ \texttt{in}\ e \\
\mid\ & var \\
\mid\ & \alpha_e
\end{aligned}
$$

2 Transformation Rules

In this section we'll show that the extended grammar can be embedded into the core grammar. To be a faithful embedding we need to show that the transformation rules preserve static and dynamic semantics. We say $e_1 \simeq e_2$ if and only if the following two holds:

- $\langle \mu_1, \Sigma_1, e_1, \varepsilon \rangle \longrightarrow_* \langle \mu_2, \Sigma_2, v, \varepsilon \rangle \iff \langle \mu_1, \Sigma_1, e_2, \varepsilon \rangle \longrightarrow_* \langle \mu_2, \Sigma_2, v, \varepsilon \rangle$
- $e_1 : \tau\ \texttt{with}\ \varepsilon$ and $e_2 : \tau\ \texttt{with}\ \varepsilon$

$\boxed{e_1 \simeq e_2}$

$$\frac{\Gamma \vdash e_1 : \tau_1 \quad \Gamma \vdash e_2 : \tau_2}{\texttt{let}\ y = e_1\ \texttt{in}\ e_2 \simeq (\texttt{new}\ x \Rightarrow \texttt{def}\ m(y : \tau_1) : \tau_2 = e_2).m(e_2)} \ (\simeq\text{-}\textsc{Let})$$

$$\frac{\Gamma \vdash e : \tau'}{f = \lambda x : \tau . e \simeq f = \texttt{new}\ x \Rightarrow\ \texttt{def}\ m(x : \tau) : \tau' = e} \ (\simeq\text{-}\textsc{Def}\lambda)$$

$$\frac{}{f y \simeq e[y/x]} \ (\simeq\text{-}\textsc{Apply}\lambda)$$

$$\frac{}{\texttt{val}\ var : \tau = e \simeq \texttt{let}\ \alpha_{var} = (\texttt{new}\ x \Rightarrow \texttt{def}\ m() : \tau = e)\ \texttt{in}\ e} \ (\simeq\text{-}\textsc{DefVal})$$

$$\frac{}{var \simeq \alpha_{var}.m()} \ \simeq\text{-}\textsc{ApplyVal}$$

Notes:

- α_{var} is used to denote a variable name whose value depends on var.