1 Typed λ_{lvar} calculus

Given a set D, let $\mathbb{D}$ be a 4-tuple $(D, \sqcup_D, \bot_D, \top_D)$, and let there be a function $incomp(d) := \forall d' \in D.d' \neq d \wedge d \sqcup d' = \top_D$ that models $\sqcup$-incompatibility among elements of $\mathbb{J}$, i.e $\mathbb{J} = \{d \in D | incomp(d)\}$.

1.1 Syntax

Types and environments

$$
\begin{array}{rcll}
T, U & := & \mathbf{1} & \text{unit} \\
 & | & T \times U & \text{product} \\
 & | & T \to U & \lambda \text{ abstraction} \\
 & | & \mathcal{J} & \text{threshold set, where } \mathcal{J} \subseteq \mathbb{J} \\
 & | & \mathcal{D}^d & \text{values } d \text{ in } D \text{ indexed by } \bigsqcup d \\
 & | & \mathcal{L}^d_{\mathcal{D}} & \text{locations (indexed by } d\text{) of values in } \mathcal{D}^d
\end{array}
$$

$$
\begin{array}{rcll}
\Gamma & := & \cdot & \text{empty environment} \\
 & | & x : T & \text{environment extension}
\end{array}
$$

Terms and Status bits

$$
\begin{array}{rcll}
L, M, N & := & x & \text{variables} \\
 & | & V & \text{values} \\
 & | & B & \text{status bits} \\
 & | & K & \text{constants} \\
 & | & M\ N & \text{parallel application} \\
 & | & () & \text{unit introduction} \\
 & | & \mathbf{let}\ () = M\ \mathbf{in}\ N & \text{unit elimination} \\
 & | & (M, N) & \text{product introduction} \\
 & | & \mathbf{let}\ (x, y) = M\ \mathbf{in}\ N & \text{product elimination}
\end{array}
$$

$$
\begin{array}{rcll}
B & := & 1 & \text{frozen LVar} \\
 & | & 0 & \text{unfrozen LVar}
\end{array}
$$

Stores and Configurations

$$
\begin{array}{rcll}
S & := & \cdot & \text{empty store} \\
 & | & S, l \mapsto (B, V) & \text{store extension}
\end{array}
$$

$$
\begin{array}{rcll}
C & := & \langle M \mid S \rangle & \text{programs are pairs of terms and store}
\end{array}
$$

Values

$$
\begin{array}{rcll}
V, W & \coloneqq & l & \text{locations} \\
 & \mid & J & \text{threshold set} \\
 & \mid & D & \text{distinguished set} \\
 & \mid & \lambda x.M & \lambda \text{ abstraction} \\
 & \mid & () & \text{unit} \\
 & \mid & (M, N) & \text{product}
\end{array}
$$

Constants

$$
\begin{array}{rcll}
K & \coloneqq & \textbf{new} & \text{allocate new LVar} \\
 & \mid & \textbf{freeze} & \text{freeze LVar} \\
 & \mid & \textbf{get} & \text{read threshold from LVar} \\
 & \mid & \textbf{put} & \text{add value to LVar}
\end{array}
$$

Evaluation Context

$$
\begin{array}{rcl}
E & \coloneqq & \square \\
 & \mid & V\ E \\
 & \mid & E\ V \\
 & \mid & (V, E) \\
 & \mid & (E, V) \\
 & \mid & \textbf{let } () = E \textbf{ in } M \\
 & \mid & \textbf{let } (x, y) = E \textbf{ in } M
\end{array}
$$

1.2 Type System

$$
\frac{}{x : T \vdash x : T}\ \textsc{T-Var}
\qquad
\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x.M : T \to U}\ \textsc{T-Lam}
\qquad
\frac{\Gamma \vdash M : T \to U \qquad \Gamma \vdash N : T}{\Gamma \vdash M\ N : U}\ \textsc{T-App}
$$

$$
\frac{}{\cdot \vdash () : \mathbf{1}}\ \textsc{T-Unit}
\qquad
\frac{\Gamma \vdash M : \mathbf{1} \qquad \Gamma \vdash N : T}{\Gamma \vdash \textbf{let } () = M \textbf{ in } N : T}\ \textsc{T-LetUnit}
$$

$$
\frac{\Gamma \vdash M : T \qquad \Gamma \vdash N : U}{\Gamma \vdash (M, N) : T \times U}\ \textsc{T-Pair}
\qquad
\frac{\Gamma \vdash M : T \times T' \qquad \Gamma, x : T, y : T' \vdash N : U}{\Gamma \vdash \textbf{let } (x, y) = M \textbf{ in } N : U}\ \textsc{T-LetPair}
$$

$$
\frac{}{\Gamma, l : \mathcal{L}_{\mathcal{D}}^{\bot} \vdash \textbf{new} : \mathcal{L}_{\mathcal{D}}^{\bot}}\ \textsc{T-New}
\qquad
\frac{\Gamma \vdash l : \mathcal{L}_{\mathcal{D}}^{d}}{\Gamma \vdash \textbf{freeze}\ l : \mathcal{D}^{d}}\ \textsc{T-Freeze}
$$

$$
\frac{\Gamma \vdash l : \mathcal{L}_{\mathcal{D}}^{d} \qquad \Gamma \vdash J : \mathcal{J} \qquad \exists d' \in \mathcal{J}.d \sqsubseteq d'}{\Gamma \vdash \textbf{get}\ l\ J : \mathcal{D}^{d}}\ \textsc{T-Get}
\qquad
\frac{\Gamma \vdash l : \mathcal{L}_{\mathcal{D}}^{d} \qquad \Gamma \vdash M : \mathcal{D}^{d'}}{\Gamma, l : \mathcal{L}_{\mathcal{D}}^{d \sqcup d'} \vdash \textbf{put}\ l\ M : 1}\ \textsc{T-Put}
$$

1.3 Operational semantics

Configuration-independent reductions

$$
\begin{array}{llll}
\textsc{E-Lam} & (\lambda x.M)\ V & \longrightarrow_M & M\{V/x\} \\
\textsc{E-Unit} & \mathbf{let}\ () = ()\ \mathbf{in}\ M & \longrightarrow_M & M \\
\textsc{E-Pair} & \mathbf{let}\ (x, y) = (V, W)\ \mathbf{in}\ M & \longrightarrow_M & M\{V/x\}\{W/y\}
\end{array}
$$

Configuration-dependent reductions

$$
\begin{array}{llll}
\textsc{E-New} & \langle E[\mathbf{new}] \mid S \rangle & \longrightarrow & \langle E[l] \mid S, l \mapsto (0, \bot) \rangle \\
\textsc{E-Freeze} & \langle E[\mathbf{freeze}] \mid S, l \mapsto (b, d) \rangle & \longrightarrow & \langle E[d] \mid S, l \mapsto (1, d) \rangle \\
\textsc{E-Put} & \langle E[\mathbf{put}\ l\ d] \mid S, l \mapsto (0, d') \rangle & \longrightarrow & \langle E[()] \mid S, l \mapsto (0, d' \sqcup d) \rangle
\end{array}
$$

$$
\textsc{E-Get}\quad \frac{J \in \mathcal{J} \qquad d' \in J \qquad d' \sqsubseteq d}{\langle E[\mathbf{get}\ l\ J] \mid S, l \mapsto (b, d) \rangle \longrightarrow \langle E[d'] \mid S, l \mapsto (b, d) \rangle}
$$

$$
\textsc{E-Pair'}\quad \frac{\langle E[M] \mid S \rangle \longrightarrow \langle E[M'] \mid S' \rangle \qquad \langle E[N] \mid S \rangle \longrightarrow \langle E[N'] \mid S'' \rangle}{\langle E[(M, N)] \mid S \rangle \longrightarrow \langle E[(M', N')] \mid S' \sqcup S'' \rangle}
$$

$$
\textsc{E-App}\quad \frac{\langle E[M] \mid S \rangle \longrightarrow \langle E[M'] \mid S' \rangle \qquad \langle E[N] \mid S \rangle \longrightarrow \langle E[N'] \mid S'' \rangle}{\langle E[M\ N] \mid S \rangle \longrightarrow \langle E[M'\ N'] \mid S' \sqcup S'' \rangle}
$$

$$
\textsc{E-Lift}\quad \frac{M \longrightarrow_M M'}{\langle E[M] \mid S \rangle \longrightarrow \langle E[M'] \mid S \rangle}
$$

1.4 Syntax sugar

runLVar

$$
\textsc{T-RunLVar}\quad \frac{\Gamma \vdash M : \mathcal{L}^d_{\mathcal{D}} \to ()}{\Gamma \vdash \mathbf{runLVar}\ M : \mathcal{D}^d}
$$

$$
\textsc{E-RunLVar}\quad \mathbf{runLVar}\ M \quad \longrightarrow \quad (\lambda l.\mathbf{let}\ () = M\ l\ \mathbf{in}\ \mathbf{freeze}\ l)\ \mathbf{new}
$$

2 Metatheory of Typed λ_{lvar} calculus