Normalized Core Language

suhorng

November 24, 2014

1 Introduction

2 Syntax

$$
\begin{aligned}
v \;::=\;& () \\
\mid\;& n \in \mathbb{N} \\
\mid\;& b \in \mathbb{B} \\
\mid\;& x
\end{aligned}
\qquad
\begin{aligned}
f \;::=\;& \lambda x.e \\
\mid\;& v\ v \\
\mid\;& \mathit{ref}\ v \\
\mid\;& !v \\
\mid\;& v := v
\end{aligned}
\qquad
\begin{aligned}
e \;::=\;& v \\
\mid\;& \textbf{let}\ x = v\ \textbf{in}\ e \\
\mid\;& \textbf{let}\ x = f\ \textbf{in}\ e \\
\mid\;& \textbf{if}\ v\ \textbf{then}\ e_1\ \textbf{else}\ e_2
\end{aligned}
$$

3 Transforming to NCL

4 An Abstract Machine

$$
\begin{aligned}
\text{Addr} &\equiv \text{(left abstract)} \\
\text{Val} &\equiv \langle \textbf{unit} \rangle \mid \langle \textbf{int}\ \mathbb{Z} \rangle \mid \langle \textbf{bool}\ \mathbb{B} \rangle \mid \langle \textbf{ref}\ \text{Addr} \rangle \mid \langle \textbf{cl}\ \text{Var}, \text{Env}, \text{Var}, e \rangle \\
\Gamma \in \text{Env} &\equiv \text{Var} \to \text{Addr} \\
\sigma \in \text{Store} &\equiv \text{Addr} \to \text{Value} \\
\kappa \in \text{Kont} &\equiv \langle \textbf{halt} \rangle \mid \langle \textbf{frame}\ \text{Env}, \text{Var}, e, \text{Kont} \rangle
\end{aligned}
$$

4.1 computation

Let $\kappa' = \langle \mathbf{frame}\, \Gamma, y, e', \kappa \rangle$.

- $\langle \Gamma, \sigma, (\lambda x.e), y, e', \kappa \rangle \longmapsto \langle \sigma, v, \kappa' \rangle$
  - $v = \langle \mathbf{cl}\, y, \Gamma, x, e \rangle$
- $\langle \Gamma, \sigma, (v_1\ v_2), y, e', \kappa \rangle \longmapsto \langle \Gamma'', \sigma', e'', \kappa' \rangle$
  - $\langle \mathbf{cl}\, \eta, \Gamma', x', e'' \rangle = \gamma(\Gamma, \sigma, v_1)$
  - $l \notin \mathrm{dom}(\sigma)$
  - $\Gamma'' = \Gamma'[x' := l]$
  - $\sigma' = \sigma[l := \gamma(\Gamma, \sigma, v_2)]$
- $\langle \Gamma, \sigma, (\texttt{ref}\ v), y, e', \kappa \rangle \longmapsto \langle \sigma', \langle \mathbf{ref}\ l \rangle, \kappa' \rangle$
  - $l \notin \mathrm{dom}(\sigma)$
  - $\sigma' = \sigma[l := \gamma(\Gamma, \sigma, v)]$
- $\langle \Gamma, \sigma, (\texttt{!}v), y, e', \kappa \rangle \longmapsto \langle \sigma, \sigma(l), \kappa' \rangle$
  - $\langle \mathbf{ref}\ l \rangle = \gamma(\Gamma, \sigma, v)$ and $l \in \mathrm{dom}(\sigma)$
- $\langle \Gamma, \sigma, (v_1 \texttt{ := } v_2), y, e', \kappa \rangle \longmapsto \langle \sigma', \langle \mathbf{unit} \rangle, \kappa' \rangle$
  - $\langle \mathbf{ref}\ l \rangle = \gamma(\Gamma, \sigma, v_1)$ and $l \in \mathrm{dom}(\sigma)$
  - $\sigma' = \sigma[l := \gamma(\Gamma, \sigma, v_2)]$

4.2 expressions

- $\langle \Gamma, \sigma, v, \kappa \rangle \longmapsto \langle \sigma, \gamma(\Gamma, \sigma, v), \kappa \rangle$
- $\langle \Gamma, \sigma, (\texttt{let}\ x = v\ \texttt{in}\ e), \kappa \rangle \longmapsto \langle \sigma, \gamma(\Gamma, \sigma, v), \kappa' \rangle$
  - $\kappa' = \langle \mathbf{frame}\, \Gamma, x, e, \kappa \rangle$.
- $\langle \Gamma, \sigma, (\texttt{let}\ x = f\ \texttt{in}\ e), \kappa \rangle \longmapsto \langle \Gamma, \sigma, f, x, e, \kappa \rangle$
- $\langle \Gamma, \sigma, (\texttt{if}\ v\ \texttt{then}\ e_1\ \texttt{else}\ e_2), \kappa \rangle \longmapsto \langle \Gamma, \sigma, e, \kappa \rangle$
  - $e = \begin{cases} e_1, & \text{if } \gamma(\Gamma, \sigma, v) = \langle \mathbf{bool}\ \textit{true} \rangle \\ e_2, & \text{if } \gamma(\Gamma, \sigma, v) = \langle \mathbf{bool}\ \textit{false} \rangle \end{cases}$

4.3 values

- $\langle \sigma, v, \langle \mathbf{frame}\, \Gamma, x, e, \kappa \rangle \rangle \longmapsto \langle \Gamma', \sigma', e, \kappa \rangle$
  - $l \notin \mathrm{dom}(\sigma)$
  - $\Gamma' = \Gamma[x := l]$
  - $\sigma' = \sigma[l := v]$

The function $\gamma : \mathrm{Env} \times \mathrm{Store} \times v \to \mathrm{Val}$ is defined by

- $\gamma(\Gamma, \sigma, \texttt{()}) = \langle \mathbf{unit} \rangle$
- $\gamma(\Gamma, \sigma, n) = \langle \mathbf{int}\ n \rangle$
- $\gamma(\Gamma, \sigma, b) = \langle \mathbf{bool}\ b \rangle$
- $\gamma(\Gamma, \sigma, x) = \sigma(\Gamma(x))$