コード生成 + Shift0/Reset0 の型システム

大石純平

平成 28 年 7 月 29 日

answer type は考えていない.
後で，answer type を加えたやつを考える.
answer type modification については考えない

1 Syntax

$$
\begin{aligned}
v &::= c \mid \lambda x.e \mid \texttt{<}e\texttt{>} \\
e &::= x \mid c \mid \lambda x.e \mid e\ e \\
&\quad \mid \underline{\lambda} x.e \mid \textbf{reset0}\ e \mid \textbf{shift0}\ k \to e \mid \textbf{throw}\ k\ e \\
&\quad \mid \underline{\textbf{clet}}\ x = e\ \underline{\textbf{in}}\ e \mid \cdots \\
c &::= N \mid B \mid \% \mid \underline{@}\ \mid + \mid \underline{+}
\end{aligned}
$$

N is Integer numeric, B is Bool (true or false)

2 Semantics

left-to-right, call-by-value

2.1 Evaluation Context

$$E ::= [\ ] \mid E\ e \mid v\ E \mid \textbf{reset0}\ E \mid \underline{\lambda} x.E$$

2.2 Operation Semantics

underline 付きのものは，コードコンビネータであり，なにか値を受け取ってコードを出すもの
underline がないもの： present stage で動く
underline があるもの： present stage で動かない
shift0 reset0 throw は コードの型を持つ e のみを引数に取ることにする？
コードレベルで shift0/reset0 throw は出てこないようにする？

throw $k\ e$ ってあるけど，これ，**throw** e にしたほうがいいな．

$$
\begin{aligned}
E[(\lambda x.e)\ v] &\rightsquigarrow E[e\{x := v\}] \\
E[\textbf{reset0}\ \texttt{<}e\texttt{>}] &\rightsquigarrow E[\texttt{<}e\texttt{>}] \\
E[\underline{\lambda} x.e] &\rightsquigarrow E[\underline{\lambda} y.e\{x := \texttt{<}y\texttt{>}\}] \quad y \text{ is fresh variable} \\
E[\underline{\lambda} y.\texttt{<}e\texttt{>}] &\rightsquigarrow E[\texttt{<}\lambda y.e\texttt{>}] \\
E[\textbf{reset0}(E'[\textbf{shift0}\ k \to E''[\textbf{throw}\ k\ e]])] &\rightsquigarrow E[E''[(k\ e)\{k := \underline{\lambda} x.\textbf{reset0}\ (E'[x])\}]] \\
& \quad x \text{ is fresh variable} \\
E[\texttt{<}e_1\texttt{>}\ \underline{@}\ \texttt{<}e_2\texttt{>}] &\rightsquigarrow E[\texttt{<}e_1\ e_2\texttt{>}] \\
E[\underline{\textbf{clet}}\ x = e_1\ \underline{\textbf{in}}\ e_2] &\rightsquigarrow E[\underline{\lambda} x.e_2\ \underline{@}\ e_1] \\
E[\%n] &\rightsquigarrow E[\texttt{<}n\texttt{>}] \\
E[\texttt{<}e_1\texttt{>}\ \underline{+}\ \texttt{<}e_2\texttt{>}] &\rightsquigarrow E[\texttt{<}e_1 + e_2\texttt{>}]
\end{aligned}
$$

3 Type System

$$t ::= \text{BasicType} \mid t \to t \mid \langle t \rangle^{\gamma}$$

Typing rule for code-level lambda:

$$\frac{\Gamma,\ \gamma_1 \geq \gamma,\ x : \langle t_1 \rangle^{\gamma_1} \vdash e\ :\ \langle t_2 \rangle^{\gamma_1}}{\Gamma \vdash \underline{\lambda} x.e\ :\ \langle t_1 \to t_2 \rangle^{\gamma}}\ (\gamma_1 \text{ is eigen var})$$

Typing rule for code-level let (derived rule):

$$\frac{\Gamma \vdash e_1\ :\ \langle t_1 \rangle^{\gamma} \quad \Gamma,\ \gamma_1 \geq \gamma,\ x : \langle t_1 \rangle^{\gamma_1} \vdash e_2\ :\ \langle t_2 \rangle^{\gamma_1}}{\Gamma \vdash \underline{\textbf{clet}}\ x = e_1\ \underline{\textbf{in}}\ e_2\ :\ \langle t_2 \rangle^{\gamma}}\ (\gamma_1 \text{ is eigen var})$$

reset0, shift0, throw のアンダーラインは取る？→ present stage で shift reset throw も動くので．
Typing rule for code-level reset0:

$$\frac{\Gamma \vdash e\ :\ \langle t \rangle^{\gamma}}{\Gamma \vdash \underline{\textbf{reset0}}\ e\ :\ \langle t \rangle^{\gamma}}$$

Typing rule for code-level shift0:

$$\frac{\Gamma,\ k : (\langle t_1 \rangle^{\gamma_1} \Rightarrow \langle t_0 \rangle^{\gamma_0}) \vdash e\ :\ \langle t_0 \rangle^{\gamma_0} \quad \Gamma \models \gamma_1 \geq \gamma_0}{\Gamma \vdash \underline{\textbf{shift0}}\ k \to e\ :\ \langle t_1 \rangle^{\gamma_1}}$$

Typing rule for code-level throw:

$$\frac{\Gamma,\ \gamma_3 \geq \gamma_1,\ \gamma_3 \geq \gamma_2 \vdash e\ :\ \langle t_1 \rangle^{\gamma_3} \quad \Gamma \models \gamma_2 \geq \gamma_0}{\Gamma,\ k : (\langle t_1 \rangle^{\gamma_1} \Rightarrow \langle t_0 \rangle^{\gamma_0}) \vdash \underline{\textbf{throw}}\ k\ e\ :\ \langle t_0 \rangle^{\gamma_2}}\ (\gamma_3 \text{ is eigen var})$$

4 Example

$$
\begin{aligned}
e_1 = \textbf{reset0}\ \ & \underline{\textbf{clet}}\ x_1 = \%3\ \underline{\textbf{in}} \\
& \textbf{reset0}\ \ \underline{\textbf{clet}}\ x_2 = \%5\ \underline{\textbf{in}} \\
& \textbf{shift0}\ k\ \to\ \underline{\textbf{clet}}\ y = t\ \underline{\textbf{in}} \\
& \textbf{throw}\ k\ (x_1\ \underline{+}\ x_2\ \underline{+}\ y)
\end{aligned}
$$

If $t = \%7$ or $t = x_1$, then e_1 is typable.

If $t = x_2$, then e_1 is not typable.

$$
\begin{aligned}
e_2 = \; & \mathbf{reset0}\ \ \underline{\mathbf{clet}}\ x_1 = \%3\ \underline{\mathbf{in}} \\
& \mathbf{reset0}\ \ \underline{\mathbf{clet}}\ x_2 = \%5\ \underline{\mathbf{in}} \\
& \mathbf{shift0}\ k_2\ \rightarrow\ \mathbf{shift0}\ k_1\ \rightarrow\ \underline{\mathbf{clet}}\ y = t\ \underline{\mathbf{in}} \\
& \mathbf{throw}\ k_1\ (\mathbf{throw}\ k_2\ (x_1\ \underline{+}\ x_2\ \underline{+}\ y))
\end{aligned}
$$

If $t = \%7$, then e_1 is typable.
If $t = x_2$ or $t = x_1$, then e_1 is not typable.