Higher-Ranked Exception Types

Ruud Koot

May 13, 2014

1 The $\lambda^{\cup}$-calculus

Types

$$\begin{array}{rcll} \tau \in \mathbf{Ty} & ::= & \mathcal{P} & \text{(base type)} \\ & | & \tau_1 \to \tau_2 & \text{(function type)} \end{array}$$

Terms

$$\begin{array}{rcll} t \in \mathbf{Tm} & ::= & x, y, ... & \text{(variable)} \\ & | & \lambda x : \tau . t & \text{(abstraction)} \\ & | & t_1\ t_2 & \text{(application)} \\ & | & \varnothing & \text{(empty)} \\ & | & \{c\} & \text{(singleton)} \\ & | & t_1 \cup t_2 & \text{(union)} \end{array}$$

Values Values v are terms of the form

$$\lambda x_1 : \tau_1 . \cdots \lambda x_i : \tau_i . \{c_1\} \cup (\cdots \cup (\{c_j\} \cup (x_1\ v_{11} \cdots v_{1m} \cup (\cdots \cup x_k\ v_{k1} \cdots v_{kn}))))$$

Environments

$$\Gamma \in \mathbf{Env} ::= \quad \cdot \quad | \quad \Gamma, x : \tau$$

1.1 Typing relation

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ [\text{T-Var}] \quad \frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash \lambda x : \tau_1 . t : \tau_1 \to \tau_2}\ [\text{T-Abs}] \quad \frac{\Gamma \vdash t_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1\ t_2 : \tau_2}\ [\text{T-App}]$$

$$\frac{}{\Gamma \vdash \varnothing : \mathcal{P}}\ [\text{T-Empty}] \quad \frac{}{\Gamma \vdash \{c\} : \mathcal{P}}\ [\text{T-Con}] \quad \frac{\Gamma \vdash t_1 : \tau \quad \Gamma \vdash t_2 : \tau}{\Gamma \vdash t_1 \cup t_2 : \tau}\ [\text{T-Union}]$$

1.2 Semantics

1.3 Reduction relation

Definition 1. *Let $\prec$ be a strict total order on $\mathbf{Con} \cup \mathbf{Var}$, with $c \prec x$ for all $c \in \mathbf{Con}$ and $x \in \mathbf{Var}$.*

$$
\begin{aligned}
(\lambda x:\tau.t_1)\ \ t_2 &\longrightarrow [t_2/x]\,t_1 && && (\beta\text{-reduction})\\
(t_1 \cup t_2)\ \ t_3 &\longrightarrow t_1\ t_3 \cup t_2\ t_3 && &&\\
(\lambda x:\tau.t_1) \cup (\lambda x:\tau.t_2) &\longrightarrow \lambda x:\tau.\,(t_1 \cup t_2) && && \text{(congruences)}\\
x\ t_1 \cdots t_n \cup x'\ t'_1 \cdots t'_n &\longrightarrow x\ \ (t_1 \cup t'_1) \cdots (t_n \cup t'_n) && &&\\
(t_1 \cup t_2) \cup t_3 &\longrightarrow t_1 \cup (t_2 \cup t_3) && && \text{(associativity)}\\
\varnothing \cup t &\longrightarrow t && && \\
t \cup \varnothing &\longrightarrow t && && \text{(unit)}\\
x \cup x &\longrightarrow x && &&\\
x \cup (x \cup t) &\longrightarrow x \cup t && &&\\
\{c\} \cup \{c\} &\longrightarrow \{c\} && && \text{(idempotence)}\\
\{c\} \cup (\{c\} \cup t) &\longrightarrow \{c\} \cup t && &&\\
x\ t_1 \cdots t_n \cup \{c\} &\longrightarrow \{c\} \cup x\ t_1 \cdots t_n && && (1)\\
x\ t_1 \cdots t_n \cup (\{c\} \cup t) &\longrightarrow \{c\} \cup (x\ t_1 \cdots t_n \cup t) && && (2)\\
x\ t_1 \cdots t_n \cup x'\ t'_1 \cdots t'_n &\longrightarrow x'\ t'_1 \cdots t'_n \cup x\ t_1 \cdots t_n && \text{if } x' \prec x && (3)\\
x\ t_1 \cdots t_n \cup (x'\ t'_1 \cdots t'_n \cup t) &\longrightarrow x'\ t'_1 \cdots t'_n \cup (x\ t_1 \cdots t_n \cup t) && \text{if } x' \prec x && (4)\\
\{c\} \cup \{c'\} &\longrightarrow \{c'\} \cup \{c\} && \text{if } c' \prec c && (5)\\
\{c\} \cup (\{c'\} \cup t) &\longrightarrow \{c'\} \cup (\{c\} \cup t) && \text{if } c' \prec c && (6)
\end{aligned}
$$

Conjecture 1. *The reduction relation $\longrightarrow$ preserves meaning.*

Conjecture 2. *The reduction relation $\longrightarrow$ is strongly normalizing.*

Conjecture 3. *The reduction relation $\longrightarrow$ is locally confluent.*

Corollary 1. *The reduction relation $\longrightarrow$ is confluent.*

Proof. Follows from SN, LC and Newman's Lemma. □

Corollary 2. *The $\lambda^{\cup}$-calculus has unique normal forms.*

Corollary 3. *Equality of $\lambda^{\cup}$-terms can be decided by normalization.*

2 Completion

$$
\begin{aligned}
\kappa \in \textbf{Kind} \quad &::= \quad \textsc{exn} && \text{(exception)}\\
&\quad\ \mid \quad \kappa_1 \Rightarrow \kappa_2 && \text{(exception operator)}\\
\chi \in \textbf{Exn} \quad &::= \quad e && \text{(exception variables)}\\
&\quad\ \mid \quad \lambda e:\kappa.\chi && \text{(exception abstraction)}\\
\widehat{\tau} \in \textbf{ExnTy} \quad &::= \quad \forall e :: \kappa.\widehat{\tau} && \text{(exception quantification)}\\
&\quad\ \mid \quad \widehat{\text{bool}} && \text{(boolean type)}\\
&\quad\ \mid \quad [\widehat{\tau}\ \textbf{throws}\ \chi] && \text{(list type)}\\
&\quad\ \mid \quad \widehat{\tau}_1\ \textbf{throws}\ \chi_1 \rightarrow \widehat{\tau}_2\ \textbf{throws}\ \chi_2 && \text{(function type)}
\end{aligned}
$$

The completion procedure as a set of inference rules:
The completion procedure as an algorithm:

$complete :: \textbf{Env} \times \textbf{Ty} \rightarrow \textbf{ExnTy} \times \textbf{Exn} \times \textbf{Env}$
$complete\ \overline{e_i :: \kappa_i}\ \textbf{bool} =$
  let *e be fresh*
  in $\langle \widehat{\text{bool}}; e\ \overline{e_i}; e :: \overline{\kappa_i \Rightarrow}\ \textsc{exn} \rangle$

$$\frac{}{\overline{e_i :: \kappa_i} \vdash \mathbf{bool} : \widehat{\mathrm{bool}}\ \&\ e\ \overline{e_i} \triangleright e :: \overline{\kappa_i} \Rightarrow \textsc{exn}}\ [\text{C-Bool}]$$

$$\frac{\overline{e_i :: \kappa_i} \vdash \tau : \widehat{\tau}\ \&\ \chi \triangleright \overline{e_j :: \kappa_j}}{\overline{e_i :: \kappa_i} \vdash [\tau] : [\widehat{\tau}\ \mathbf{throws}\ \chi]\ \&\ e\ \overline{e_i} \triangleright e :: \overline{\kappa_i} \Rightarrow \textsc{exn}, \overline{e_j :: \kappa_j}}\ [\text{C-List}]$$

$$\frac{\vdash \tau_1 : \widehat{\tau}_1\ \&\ \chi_1 \triangleright \overline{e_j :: \kappa_j} \quad \overline{e_i :: \kappa_i}, \overline{e_j :: \kappa_j} \vdash \tau_2 : \widehat{\tau}_2\ \&\ \chi_2 \triangleright \overline{e_j :: \kappa_j}}{\overline{e_i :: \kappa_i} \vdash \tau_1 \to \tau_2 : \forall \overline{e_j :: \kappa_j}. \left(\widehat{\tau}_1\ \mathbf{throws}\ \chi_1 \to \widehat{\tau}_2\ \mathbf{throws}\ \chi_2\right)\ \&\ e\ \overline{e_i} \triangleright e :: \overline{\kappa_j} \Rightarrow \textsc{exn}, \overline{e_k :: \kappa_k}}\ [\text{C-Arr}]$$

Figure 1: Type completion ($\Gamma \vdash \tau : \widehat{\tau}\ \&\ \chi \triangleright \Gamma'$)