

Higher-Ranked Exception Types

Ruud Koot

April 4, 2014

1 The λ^U -calculus

Types

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

Terms

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

Values Values v are terms of the form

$$\lambda x_1 : \tau_1. \dots \lambda x_i : \tau_i. \{c_1\} \cup (\dots \cup (\{c_j\} \cup (x_1 v_{11} \dots v_{1m} \cup (\dots \cup x_k v_k 1 \dots v_k n))))).$$

Environments

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

1.1 Typing relation

$$\begin{array}{lll} \frac{}{\Gamma, x : \tau \vdash x : \tau} [\mathbf{T-VAR}] & \frac{\Gamma, x : \tau_1 \vdash t : \tau_2}{\Gamma \vdash \lambda x : \tau_1. t : \tau_1 \rightarrow \tau_2} [\mathbf{T-ABS}] & \frac{\Gamma \vdash t_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash t_2 : \tau_1}{\Gamma \vdash t_1 t_2 : \tau_2} [\mathbf{T-APP}] \\ \\ \frac{}{\Gamma \vdash \emptyset : \mathcal{P}} [\mathbf{T-EMPTY}] & \frac{}{\Gamma \vdash \{c\} : \mathcal{P}} [\mathbf{T-CON}] & \frac{\Gamma \vdash t_1 : \tau \quad \Gamma \vdash t_2 : \tau}{\Gamma \vdash t_1 \cup t_2 : \tau} [\mathbf{T-UNION}] \end{array}$$

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 $x \in \mathbf{Var}$.

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

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 λ^\cup -calculus has unique normal forms.*

Corollary 3. *Equality of λ^\cup -terms can be decided by normalization.*

2 Completion

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

The completion procedure as a set of inference rules:

The completion procedure as an algorithm:

```

complete :: Env × Ty → ExnTy × Exn × Env
complete  $\bar{e}_i :: \bar{\kappa}_i$  bool =
  let  $e$  be fresh
  in  $\langle \widehat{\text{bool}}; e \ \bar{e}_i; e :: \bar{\kappa}_i \Rightarrow \text{EXN} \rangle$ 

```

$$\begin{array}{c}
\overline{e_i :: \kappa_i} \vdash \text{bool} : \widehat{\text{bool}} \ \& \ e \ \overline{e_i} \triangleright e :: \overline{\kappa_i} \Rightarrow_{\text{EXN}} \quad [\text{C-Bool}] \\
\\
\frac{\overline{e_i :: \kappa_i} \vdash \tau : \widehat{\tau} \ \& \ \chi \triangleright \overline{e_j :: \kappa_j}}{\overline{e_i :: \kappa_i} \vdash [\tau] : [\widehat{\tau} \ \text{throws} \ \chi] \ \& \ e \ \overline{e_i} \triangleright e :: \overline{\kappa_i} \Rightarrow_{\text{EXN}}, \overline{e_j :: \kappa_j}} \quad [\text{C-List}] \\
\\
\frac{\vdash \tau_1 : \widehat{\tau_1} \ \& \ \chi_1 \triangleright \overline{e_j :: \kappa_j} \quad \overline{e_i :: \kappa_i}, \overline{e_j :: \kappa_j} \vdash \tau_2 : \widehat{\tau_2} \ \& \ \chi_2 \triangleright \overline{e_j :: \kappa_j}}{\overline{e_i :: \kappa_i} \vdash \tau_1 \rightarrow \tau_2 : \forall \overline{e_j :: \kappa_j}. (\widehat{\tau_1} \ \text{throws} \ \chi_1 \rightarrow \widehat{\tau_2} \ \text{throws} \ \chi_2) \ \& \ e \ \overline{e_i} \triangleright e :: \overline{\kappa_i} \Rightarrow_{\text{EXN}}, \overline{e_j :: \kappa_j}} \quad [\text{C-Arr}]
\end{array}$$

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