Higher-Ranked Exception Types

Ruud Koot

April 2, 2014

1 The $\lambda^\cup$-calculus

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

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

Corollary 1. *The $\lambda^\cup$-calculus has unique normal forms.*

Corollary 2. *Equality of $\lambda^\cup$-terms can be decided by normalization.*

2 Completion

$$\begin{array}{llll} \kappa \in \mathbf{Kind} & ::= & \textsc{exn} & \text{(exception)} \\ & | & \kappa_1 \Rightarrow \kappa_2 & \text{(exception operator)} \end{array}$$

$$\begin{array}{llll} \chi \in \mathbf{Exn} & ::= & e & \text{(exception variables)} \\ & | & \lambda e : \kappa.\chi & \text{(exception abstraction)} \end{array}$$

$$\begin{array}{llll} \widehat{\tau} \in \mathbf{ExnTy} & ::= & \forall e :: \kappa.\widehat{\tau} & \text{(exception quantification)} \\ & | & \widehat{\mathrm{bool}} & \text{(boolean type)} \\ & | & [\widehat{\tau}\ \mathbf{throws}\ \chi] & \text{(list type)} \\ & | & \widehat{\tau}_1\ \mathbf{throws}\ \chi_1 \to \widehat{\tau}_2\ \mathbf{throws}\ \chi_2 & \text{(function type)} \end{array}$$

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

$$\begin{array}{l} \mathit{complete} :: \mathbf{Env} \times \mathbf{Ty} \to \mathbf{ExnTy} \times \mathbf{Exn} \times \mathbf{Env} \\ \mathit{complete}\ \overline{e_i :: \kappa_i}\ \mathrm{bool} = \\ \quad \mathbf{let}\ e\ \mathit{be\ fresh} \\ \quad \mathbf{in}\ \ \langle \widehat{\mathrm{bool}}; e\ \overline{e_i}; e :: \overline{\kappa_i \Rightarrow}\ \textsc{exn} \rangle \end{array}$$

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

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

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

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