Type and Effect Systems

Pepijn Kokke, Wout Elsinghorst

Dept. of Homeland Security kingofthehill@cs.uu.nl

Revision: no revision

1 Introduction

Join the army! They said.
There can be only one.
It is a good day to die
Make it so.
iddqd
Pot of gold!
idkfa
See the world! They said.
Finding nemo.

2 Motivation

Our original motivation for this blasfamous work was to obtain a grade. Further motivation can be found under your chair.

3 Preliminaries

As you should know by now, type systems rule. Effect systems rule even more so.

4 Syntax

u	$\in$	**Unicode**	Unicode characters
z	$\in$	$\mathbb{Z}$	integers
r	$\in$	$\mathbb{R}$	reals
D	$\in$	**Con**	data constructors

Equivalence $\boxed{\varrho \equiv \varrho'}$

$$\frac{}{\varrho \equiv \varrho}\,[\textit{q-refl}] \qquad \frac{\varrho' \equiv \varrho}{\varrho \equiv \varrho'}\,[\textit{q-symm}] \qquad \frac{\varrho \equiv \varrho'' \quad \varrho'' \equiv \varrho'}{\varrho \equiv \varrho'}\,[\textit{q-trans}]$$

Figure 1: Definitional equivalence of ...

Refined typing $\boxed{\widehat{\Gamma} \vdash e :: \widehat{\tau}@\varrho}$

$$\frac{}{\widehat{\Gamma} \vdash u :: \mathsf{Char}@\{\,u\,\}}\,[\textit{r-chr}] \qquad \frac{}{\widehat{\Gamma} \vdash z :: \mathsf{Integer}@\{\,z\,\}}\,[\textit{r-int}] \qquad \frac{}{\widehat{\Gamma} \vdash r :: \mathsf{Double}@\{\,r\,\}}\,[\textit{r-flt}]$$

$$\frac{\widehat{\Gamma}(D) = \widehat{\tau}}{\widehat{\Gamma} \vdash D :: \widehat{\tau}@\{\,D\,\}}\,[\textit{r-con}] \qquad \frac{\begin{array}{c}\widehat{\Gamma} \vdash e_1 :: \mathsf{Bool}@(\{\,\mathit{True}\,\} \cup \{\,\mathit{False}\,\}) \\ \widehat{\Gamma} \vdash e_2 :: \widehat{\tau}@\varrho \quad \widehat{\Gamma} \vdash e_3 :: \widehat{\tau}@\varrho\end{array}}{\widehat{\Gamma} \vdash \mathbf{if}\ e_1\ \mathbf{then}\ e_2\ \mathbf{else}\ e_3 :: \widehat{\tau}@\varrho}\,[\textit{r-if}]$$

$$\frac{\widehat{\Gamma} \vdash e :: \widehat{\tau}@\varrho' \quad \varrho' \subseteq \varrho}{\widehat{\Gamma} \vdash e :: \widehat{\tau}@\varrho}\,[\textit{r-sub}]$$

Figure 2: Static semantics.

$$\begin{array}{lcll} l & \in & \mathbf{Literal} & \text{literals} \\ e & \in & \mathbf{Exp} & \text{expressions} \end{array}$$

$$\begin{array}{lcl} l & ::= & u \mid z \mid r \\ e & ::= & l \mid D \mid \mathbf{if}\ e_1\ \mathbf{then}\ e_2\ \mathbf{else}\ e_3 \end{array}$$

5 Type system

Well typed programs can't go wrong. Untyped lambda calculus is never wrong.

6 Algorithm

Using Quantum Mechanics and other obfuscation techniques, our algorithm has attained a form of True Elegance. No further explanation necessary.

7 Related work

It might be the case that our fellow students duplicated our work. Their results might prove fruitful.

$$
\begin{array}{l}
\mathbf{E}(\widehat{\Gamma}, l) \quad = \mathbf{L}(l) \\
\mathbf{E}(\widehat{\Gamma}, D) \quad = \text{let } \zeta \text{ be fresh in } (\widehat{\Gamma}(D)\ ,\ \zeta\ ,\ \{\{D\} \subseteq \zeta\}) \\
\mathbf{E}(\widehat{\Gamma}, \textbf{if } e_1 \textbf{ then } e_2 \textbf{ else } e_3) = \\
\quad \text{let } \zeta \text{ be fresh} \\
\qquad (\widehat{\tau}_1, \zeta_1, C_1) = \mathbf{E}(\widehat{\Gamma}, e_1) \\
\qquad (\widehat{\tau}_2, \zeta_2, C_2) = \mathbf{E}(\widehat{\Gamma}, e_2) \\
\qquad (\widehat{\tau}_3, \zeta_3, C_3) = \mathbf{E}(\widehat{\Gamma}, e_3) \\
\quad \text{in if } \quad \widehat{\tau}_1 = \mathsf{Bool} \wedge \widehat{\tau}_2 = \widehat{\tau}_3 \\
\qquad \text{then } (\widehat{\tau}_2\ ,\ \zeta\ ,\ C_1 \cup C_2 \cup C_3 \cup \{(\{\mathit{True}\} \cup \{\mathit{False}\}) \supseteq \zeta_1\} \cup \{\zeta_2 \subseteq \zeta\} \cup \{\zeta_3 \subseteq \zeta\}) \\
\qquad \text{else } \ \text{fail}
\end{array}
$$

Figure 3: Reconstruction algorithm: expressions.

$$
\begin{array}{l}
\mathbf{solve}(C) = \text{do} \\
\quad \cdots \\
\quad \text{for all } c \text{ in } C \text{ do } \mathsf{worklist} := \mathsf{worklist} \cup \{c\} \\
\quad \cdots \\
\quad \text{while } \mathsf{worklist} \neq \{\,\} \text{ do} \\
\qquad \text{let } C_1 \uplus \{c\} = \mathsf{worklist} \\
\qquad \text{in } \ \text{do} \cdots \\
\qquad\qquad \text{case } c \text{ of} \\
\qquad\qquad\quad \cdots \\
\quad \text{return } \mathsf{analysis}
\end{array}
$$

Figure 4: Worklist algorithm for constraint solving.

8 Conclusion and future work

There is not much to conclude. We rule.