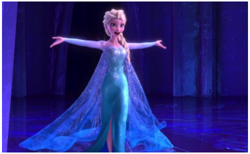
Hindley-Milner polymorphism and algebraic effects

Ohad Kammar

<ohad.kammar@cl.cam.ac.uk>

joint work with Sean Moss and Matija Pretnar



Schloss Dagstuhl Seminar:

From Theory to Practice of Algebraic Effects and Handlers

15 March 2016

UNIVERSITY OF CAMBRIDGE Computer Laboratory

Let polymorphism

$$\frac{\alpha_1, \ldots, \alpha_n, \beta_1, \ldots, \beta_m; \Gamma \vdash M : A \qquad \alpha_1, \ldots, \alpha_n \vdash \Gamma \qquad \alpha_1, \ldots, \alpha_n; \Gamma, x : \forall \beta_1, \ldots, \beta_m . A \vdash N : B}{\alpha_1, \ldots, \alpha_n; \Gamma \vdash \mathbf{let}\ x \leftarrow M\ \mathbf{in}\ N : B}$$

$$\frac{\Gamma(x) = \forall \beta_1, \ldots, \beta_m . A}{x : A[B_1/\beta_1, \ldots, B_n/\beta_n]}$$

Simulating global state locally

$$H_{ST} := \textbf{handler}\ \{\ \textbf{return}\ x \mapsto \textbf{fun}\ \_ \mapsto \textbf{return}\ x$$
$$\mathrm{get}(\_; k) \mapsto \textbf{fun}\ s \mapsto k\ s\ s$$
$$\mathrm{set}(s'; k) \mapsto \textbf{fun}\ \_ \mapsto k\ ()\ s'\}$$

$$(\textbf{with}\ H_{ST}\ \textbf{handle}\ \mathrm{set}\ \textbf{true};\ \textbf{let}\ y \leftarrow \mathrm{get}\ ()\ \textbf{in}\ \textbf{return}\ y)\ \textbf{false}$$

$$\rightsquigarrow^{*}\ (\textbf{with}\ H_{ST}\ \textbf{handle}\ \textbf{let}\ y \leftarrow \mathrm{get}\ ()\ \textbf{in}\ \textbf{return}\ y)\ \textbf{true}$$

$$\rightsquigarrow^{*}\ \textbf{return true}$$

$$\rightsquigarrow^{*}\ (\textbf{fun}\ s \mapsto \textbf{with}\ H_{ST}\ \textbf{handle}\ (\textbf{fun}\ \_ \mapsto \textbf{let}\ y \leftarrow \mathrm{get}\ ()\ \textbf{in}\ \textbf{return}\ y)\ ()\ \ \textbf{true})\ \textbf{false}$$

$$\rightsquigarrow^{*}\ (\textbf{with}\ H_{ST}\ \textbf{handle}\ \textbf{let}\ y \leftarrow \textbf{return true in}\ \textbf{return}\ y)\ \textbf{true}$$

Polymorphic types

$$
\begin{aligned}
H_{ST} := \ \textbf{handler}\ \{\ & \textbf{return}\ x \mapsto \textbf{fun}\ \_ \mapsto \textbf{return}\ x \\
& \mathrm{get}(\_; k) \mapsto \textbf{fun}\ s \mapsto k\ \ s\ \ s \\
& \mathrm{set}(s'; k) \mapsto \textbf{fun}\ \_ \mapsto k\ ()\ s' \}
\end{aligned}
$$

$$H_{ST} : \forall \alpha, \beta . \alpha \,!\{\mathrm{get} : \mathrm{unit} \to \beta, \mathrm{set} : \beta \to \mathrm{unit}\} \Rightarrow (\beta \to \alpha \,!\, \emptyset) \,!\, \emptyset$$

Untyped Eff

Syntax

value	$v ::=$	x	variable
	$\mid$	$\mathbf{true} \mid \mathbf{false}$	boolean constants
	$\mid$	$\mathbf{fun}\ x \mapsto c$	function
	$\mid$	h	handler
handler	$h ::=$	$\mathbf{handler}\ \{\mathbf{return}\ x \mapsto c_r,$	return clause
		$\mathrm{op}_1(x; k) \mapsto c_1, \ldots, \mathrm{op}_n(x; k) \mapsto c_n\}$	operation clauses
computation	$c ::=$	$\mathbf{return}\ v$	return
	$\mid$	$\mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2$	sequencing
	$\mid$	$\mathrm{op}(v; y.\, c)$	operation call
	$\mid$	$\mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2$	conditional
	$\mid$	$v_1\ v_2$	application
	$\mid$	$\mathbf{with}\ v\ \mathbf{handle}\ c$	handling

Untyped Eff

Semantics (part 1)

$$\frac{c_1 \rightsquigarrow c_1'}{\mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2 \rightsquigarrow \mathbf{let}\ x \leftarrow c_1'\ \mathbf{in}\ c_2}$$

$$\frac{}{\mathbf{let}\ x \leftarrow \mathbf{return}\ v\ \mathbf{in}\ c \rightsquigarrow c[v/x]}$$

$$\frac{}{\mathbf{if\ true\ then}\ c_1\ \mathbf{else}\ c_2 \rightsquigarrow c_1} \qquad \frac{}{\mathbf{if\ false\ then}\ c_1\ \mathbf{else}\ c_2 \rightsquigarrow c_2}$$

$$\frac{}{(\mathbf{fun}\ x \mapsto c)\ v \rightsquigarrow c[v/x]}$$

$$\frac{}{\mathbf{let}\ x \leftarrow \mathrm{op}(v; y.\, c_1)\ \mathbf{in}\ c_2 \rightsquigarrow \mathrm{op}(v; y.\, \mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2)}\,(\text{DO-OP})$$

Semantics (part 2)

For every

$h = \textbf{handler}\ \{\textbf{return}\ x \mapsto c_r, \mathrm{op}_1(x;k) \mapsto c_1, \ldots, \mathrm{op}_n(x;k) \mapsto c_n\}$,

define:

$$\frac{c \rightsquigarrow c'}{\textbf{with}\ h\ \textbf{handle}\ c \rightsquigarrow \textbf{with}\ h\ \textbf{handle}\ c'}$$

$$\frac{}{\textbf{with}\ h\ \textbf{handle}\ (\textbf{return}\ v) \rightsquigarrow c_r[v/x]}$$

$$\frac{(1 \le i \le n)}{\textbf{with}\ h\ \textbf{handle}\ \mathrm{op}_i(v; y.\, c) \rightsquigarrow c_i[v/x, (\textbf{fun}\ y \mapsto \textbf{with}\ h\ \textbf{handle}\ c)/k]}$$

$$\frac{(\mathrm{op} \notin \{\mathrm{op}_1, \ldots, \mathrm{op}_n\})}{\textbf{with}\ h\ \textbf{handle}\ \mathrm{op}(v; y.\, c) \rightsquigarrow \mathrm{op}(v; y.\, \textbf{with}\ h\ \textbf{handle}\ c)}$$

Eff types and effects

Types

value type	A, B	$::= \alpha$	type variable
		$\mid \mathrm{bool}$	boolean type
		$\mid A \to \underline{C}$	function type
		$\mid \underline{C} \Rightarrow \underline{D}$	handler type
computation type	$\underline{C}, \underline{D}$	$::= A \,!\, \Sigma$	
scheme		$\forall \vec{\alpha}.A$	
effect signatures	Σ	$::= \{\mathrm{op}_1 : A_1 \to B_1, \ldots, \mathrm{op}_n : A_n \to B_n\}$	

Kind system

Well-formed value types:

$$\frac{\alpha \in \Theta}{\Theta \vdash \alpha} \qquad \frac{}{\Theta \vdash \mathsf{bool}} \qquad \frac{\Theta \vdash A \qquad \Theta \vdash \underline{C}}{\Theta \vdash A \to \underline{C}} \qquad \frac{\Theta \vdash C \qquad \Theta \vdash \underline{D}}{\Theta \vdash \underline{C} \Rightarrow \underline{D}}$$

Well-formed effect signatures, schemes, and computation types:

$$\frac{[\Theta \vdash A_i \qquad \Theta \vdash B_i]_{1 \le i \le n}}{\Theta \vdash \{\mathsf{op}_1 : A_1 \to B_1, \ldots, \mathsf{op}_n : A_n \to B_n\}} \qquad \frac{\Theta, \vec{\alpha} \vdash A}{\Theta \vdash \forall \vec{\alpha}.A}$$

$$\frac{\Theta \vdash A \qquad \Theta \vdash \Sigma}{\Theta \vdash A\,!\,\Sigma}$$

Well-formed polymorphic and monomorphic contexts:

$$\frac{[\Theta \vdash \forall \vec{\alpha}.A]_{(x : \forall \vec{\alpha}.A) \in \Xi}}{\Theta \vdash \Xi} \qquad \frac{[\Theta \vdash A]_{(x : A) \in \Gamma}}{\Theta \vdash \Gamma}$$

Type and effect system (part 1)

Value judgements $\boxed{\Theta; \Xi; \Gamma \vdash v : A}$**, assuming** $\Theta \vdash \Xi, \Gamma, A$**:**

$$\frac{(x : A) \in \Gamma}{\Theta; \Xi; \Gamma \vdash x : A} \qquad \frac{(x : \forall \vec{\alpha}.B) \in \Xi \qquad [\Theta \vdash A_i]_{1 \le i \le |\vec{\alpha}|}}{\Theta; \Xi; \Gamma \vdash x : B[A_i/\alpha_i]_{1 \le i \le |\vec{\alpha}|}}$$

$$\frac{}{\Theta; \Xi; \Gamma \vdash \mathbf{true} : \mathrm{bool}} \qquad \frac{}{\Theta; \Xi; \Gamma \vdash \mathbf{false} : \mathrm{bool}}$$

$$\frac{\Theta; \Xi; \Gamma, x : A \vdash c : \underline{C}}{\Theta; \Xi; \Gamma \vdash \mathbf{fun}\ x \mapsto c : A \to \underline{C}}$$

$$\frac{\begin{array}{c} \Theta; \Xi; \Gamma, x : A \vdash c_r : B \,!\, \Sigma' \\ \left[(\mathrm{op}_i : A_i \to B_i) \in \Sigma \qquad \Theta; \Xi; \Gamma, x : A_i, k : B_i \to B \,!\, \Sigma' \vdash c_i : B \,!\, \Sigma'\right]_{1 \le i \le n} \\ \Sigma \setminus \{\mathrm{op}_i \mid 1 \le i \le n\} \subseteq \Sigma' \end{array}}{\Theta; \Xi; \Gamma \vdash \mathbf{handler}\ \{\mathbf{return}\ x \mapsto c_r, \mathrm{op}_1(x; k) \mapsto c_1, \ldots, \mathrm{op}_n(x; k) \mapsto c_n\} : A \,!\, \Sigma \Rightarrow B \,!\, \Sigma'}$$

Type and effect system (part 2)

Computation judgements $\boxed{\Theta; \Xi; \Gamma \vdash c : A \,!\, \Sigma}$**, assuming** $\Theta \vdash \Xi, \Gamma, A$**:**

$$\frac{\Theta; \Xi; \Gamma \vdash v : A}{\Theta; \Xi; \Gamma \vdash \mathbf{return}\ v : A \,!\, \Sigma} \qquad \frac{\Theta; \Xi; \Gamma \vdash c_1 : (\forall \vec{\alpha}. A) \,!\, \Sigma \qquad \Theta; \Xi, x : \forall \vec{\alpha}. A; \Gamma \vdash c_2 : B \,!\, \Sigma}{\Theta; \Xi; \Gamma \vdash \mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2 : B \,!\, \Sigma}$$

$$\frac{(\mathrm{op} : A_{\mathrm{op}} \to B_{\mathrm{op}}) \in \Sigma \qquad \Theta; \Xi; \Gamma \vdash v : A_{\mathrm{op}} \qquad \Theta; \Xi; \Gamma, y : B_{\mathrm{op}} \vdash c : A \,!\, \Sigma}{\Theta; \Xi; \Gamma \vdash \mathrm{op}(v; y.\, c) : A \,!\, \Sigma}$$

$$\frac{\Theta; \Xi; \Gamma \vdash v : \mathrm{bool} \qquad \Theta; \Xi; \Gamma \vdash c_1 : \underline{C} \qquad \Theta; \Xi; \Gamma \vdash c_2 : \underline{C}}{\Theta; \Xi; \Gamma \vdash \mathbf{if}\ v\ \mathbf{then}\ c_1\ \mathbf{else}\ c_2 : \underline{C}}$$

$$\frac{\Theta; \Xi; \Gamma \vdash v_1 : A \to \underline{C} \qquad \Theta; \Xi; \Gamma \vdash v_2 : A}{\Theta; \Xi; \Gamma \vdash v_1\ v_2 : \underline{C}} \qquad \frac{\Theta; \Xi; \Gamma \vdash v : \underline{C} \Rightarrow \underline{D} \qquad \Theta; \Xi; \Gamma \vdash c : \underline{C}}{\Theta; \Xi; \Gamma \vdash \mathbf{with}\ v\ \mathbf{handle}\ c : \underline{D}}$$

Type and effect system (part 3)

Scheme judgement $\boxed{\Theta; \Xi; \Gamma \vdash c : (\forall \vec{\alpha}.A) \,!\, \Sigma}$**, assuming** $\Theta \vdash \Xi, \Gamma, (\forall \vec{\alpha}.A), \Sigma$**:**

$$\frac{\Theta, \vec{\alpha}; \Xi; \Gamma \vdash c : A \,!\, \Sigma}{\Theta; \Xi; \Gamma \vdash c : (\forall \vec{\alpha}.A) \,!\, \Sigma}(\textsc{gen})$$

Safety

Theorem

If $\vdash c : A \,!\, \Sigma$ *holds, then either:*

(i) $c \rightsquigarrow c'$ *for some* $\vdash c' : A \,!\, \Sigma$*;*

(ii) $c = \mathbf{return}\ v$ *for some* $\vdash v : A$*; or*

(iii) $c = \mathrm{op}(v; y.\, c')$ *for some* $(\mathrm{op} : A_{\mathrm{op}} \to B_{\mathrm{op}}) \in \Sigma$, $\vdash v : A_{\mathrm{op}}$*,* *and* $y : B_{\mathrm{op}} \vdash c' : A \,!\, \Sigma$*.*

In particular, when $\Sigma = \emptyset$*, evaluation will not get stuck before returning a value.*

Safety

Proof sketch (formalised in Twelf):

$$
\dfrac{\dfrac{\dfrac{\begin{array}{c}\vdash (\mathrm{op} : A_{\mathrm{op}} \to B_{\mathrm{op}}) \in \Sigma \\ \vec{\alpha} \vdash v : A_{\mathrm{op}} \qquad \vec{\alpha}; y : B_{\mathrm{op}} \vdash c : A \,!\, \Sigma\end{array}}{\vec{\alpha} \vdash \mathrm{op}(v; y.\, c_1) : A!\Sigma}}{\vdash \mathrm{op}(v; y.\, c_1) : (\forall \vec{\alpha}.A) \,!\, \Sigma} \quad x : \forall \vec{\alpha}.A \vdash c_2 : B \,!\, \Sigma}{\vdash \mathbf{let}\ x \leftarrow \mathrm{op}(v; y.\, c_1)\ \mathbf{in}\ c_2 : B \,!\, \Sigma} \qquad \rightsquigarrow
$$

$$
\dfrac{\vdash (\mathrm{op} : A_{\mathrm{op}} \to B_{\mathrm{op}}) \in \Sigma \qquad \vdash v : A_{\mathrm{op}} \qquad \dfrac{\dfrac{\vec{\alpha}; y : B_{\mathrm{op}} \vdash c : A \,!\, \Sigma}{y : B_{\mathrm{op}} \vdash c : (\forall \vec{\alpha}.A) \,!\, \Sigma}}{y : B_{\mathrm{op}} \vdash \mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2 : B \,!\, \Sigma}}{\vdash \mathrm{op}(v; y.\, \mathbf{let}\ x \leftarrow c_1\ \mathbf{in}\ c_2) : B \,!\, \Sigma}
$$

Evaluation

Feature interaction

$$\begin{array}{l}\textbf{let}\ \mathrm{imp\_map} \leftarrow \textbf{fun}\ f\ xs \mapsto \\ \quad \textbf{with}\ H_{ST}\ \textbf{handle}\ (\textbf{foldl}\ \ (\textbf{fun}\ x \mapsto \mathsf{set}(f\ \ x :: \mathsf{get}\ ()))\ ()\ xs; \\ \qquad\qquad\qquad\qquad \mathrm{reverse}(\mathsf{get}\ ()) \\ \quad [\,]\ (*\ \text{initial state}\ *)\ \textbf{in}\ \ldots\end{array}$$

$$\mathrm{imp\_map} : \forall \alpha\beta.(\alpha \to \beta\,!\,\Sigma) \to (\alpha\ \mathsf{list} \to \beta\ \mathsf{list}\,!\,\Sigma)\,!\,\emptyset$$

for any Σ.

Unrestricted polymorphism

$$\textbf{let}\ id \leftarrow (\textbf{fun}\ f \mapsto f)(\textbf{fun}\ x \mapsto x)\ \textbf{in}\ \ldots$$

$$id : \forall\alpha(\alpha \to \alpha\,!\,\emptyset)$$

Images

- http://cfensi.files.wordpress.com/2014/01/frozen-let-it-go.png