Relational interpretation of algebraic effects

Wiktor Kuchta

January 17, 2024

Background: operational semantics

So far, we've formalized the semantics of effects and handlers as step relations between expressions:

$$
\begin{array}{lcl}
\mathsf{handle} & & \mathsf{handle} \\
\quad \mathsf{ask}() + 5 & \to & \quad 10 + 5 \\
\{\mathsf{ask}\ ()\ k \to k\ 10\} & & \{\mathsf{ask}\ ()\ k \to k\ 10\}
\end{array}
$$

Background: type-and-effect systems

Expressions have not only a type, but also potential effects:

$$\mathsf{ask}() + 5 : \mathsf{int}\ /\ \mathsf{ask}$$

Background: type-and-effect systems

Expressions have not only a type, but also potential effects:

$$\mathsf{ask}() + 5 : \mathsf{int} \ / \ \mathsf{ask}$$

Formally, we have rules for constructing typing judgments, e.g.,

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2 \ / \ \rho}{\Gamma \vdash \lambda x.\, e : \tau_1 \rightarrow_\rho \tau_2 \ / \ \cdot}$$

But what does $e : \tau \;/\; \rho$ mean?

Intuitively, "e may perform ρ before evaluating to a τ".

But what does $e : \tau \;/\; \rho$ mean?

Intuitively, "e may perform ρ before evaluating to a τ".
So it refers to how the program behaves – to the *semantics*.

But what does $e : \tau \mathbin{/} \rho$ mean?

Intuitively, "e may perform ρ before evaluating to a τ".
So it refers to how the program behaves – to the *semantics*.
Such program properties are undecidable.
A type system can only *approximate* the real notion.

Agenda

1. Definition of semantic typing, using the step relation $\rightarrow$.

$$\Gamma \models e : \tau \mathrel{/} \rho$$

2. Compatibility with (i.e., soundness of) syntactic typing.

$$\Gamma \vdash e : \tau \mathrel{/} \rho$$

   (*Corollary:* well-typed programs $\vdash e : \tau \mathrel{/} \cdot$ indeed terminate.)

3. Row polymorphism. Semantic equivalence.

$$\Delta ; \Gamma \models e_1 \approx e_2 : \tau \mathrel{/} \rho$$

Syntax

We extend the call-by-value λ-calculus.
Assume op ranges over a set of operation names, e.g. $\mathrm{ask}, \mathrm{pick}\ldots$

$$\begin{aligned} v, u &::= x \mid \lambda x.\, e \\ e &::= v \mid e\ e \mid op\ v \mid \textsf{handle}\ e\, \{op\ x\ k \to\ e\} \end{aligned}$$

Reduction

$$K ::= \square \mid K\ e \mid v\ K \mid \textsf{handle}\ K\ \{op\ x\ k \to e\}$$

$$(\lambda x.\, e)\ v \mapsto e[v/x]$$

$$\frac{K\ op\text{-free} \qquad v_c = \lambda z.\, \textsf{handle}\ K[z]\ \{op\ x\ k \to e_h\}}{\textsf{handle}\ K[op\ v]\ \{op\ x\ k \to e_h\} \mapsto e_h[v/x][v_c/k]}$$

$$\textsf{handle}\ v\ \{op\ x\ k \to e_h\} \mapsto v$$

$$\frac{e_1 \mapsto e_2}{K[e_1] \to K[e_2]}$$

Consider closed terms first.

$$\mathcal{V}[\![\mathsf{bool}]\!] = \{\mathsf{true}, \mathsf{false}\}$$
$$\mathcal{V}[\![\sigma \to_\rho \tau]\!] = \{\lambda x.\, e \mid \forall v \in \mathcal{V}[\![\sigma]\!].\, e[v/x] \in \mathcal{E}[\![\tau/\rho]\!]\}$$

Consider closed terms first.

$$\mathcal{V}[\![\mathsf{bool}]\!] = \{\mathsf{true}, \mathsf{false}\}$$
$$\mathcal{V}[\![\sigma \to_\rho \tau]\!] = \{\lambda x.\, e \mid \forall v \in \mathcal{V}[\![\sigma]\!].\, e[v/x] \in \mathcal{E}[\![\tau/\rho]\!]\}$$

$$\begin{aligned}
\mathcal{E}[\![\tau' \;/\; \overline{op_i : \sigma_i \Rightarrow \tau_i}]\!] = {} & \{e \mid \exists v \in \mathcal{V}[\![\tau']\!].\, e \to^* v\} \\
\cup {} & \{e \\
& \quad \mid \exists K, i, v.\, e \to^* K[op_i\ v] \\
& \quad \wedge K\ op_i\text{-free} \\
& \quad \wedge v \in \mathcal{V}[\![\sigma_i]\!] \\
& \quad \wedge \forall u \in \mathcal{V}[\![\tau_i]\!].\, K[u] \in \mathcal{E}[\![\tau' \;/\; \overline{op_i : \sigma_i \Rightarrow \tau_i}]\!]\}
\end{aligned}$$

Inductive definition, formally in set theory

$\mathcal{E}[\![\tau' \,/\, \overline{op_i : \sigma_i \Rightarrow \tau_i}]\!]$ is the least fixed point of the following function.

$$
\begin{aligned}
\mathcal{E}'(X) = {} & \{e \mid \exists v \in \mathcal{V}[\![\tau']\!].\, e \to^* v\} \\
& \cup \{e \\
& \quad \mid \exists K, i, v.\, e \to^* K[op_i\ v] \\
& \quad \wedge K\ op_i\text{-free} \\
& \quad \wedge v \in \mathcal{V}[\![\sigma_i]\!] \\
& \quad \wedge \forall u \in \mathcal{V}[\![\tau_i]\!].\, K[u] \in X\}
\end{aligned}
$$

Inductive definition, formally in set theory

$\mathcal{E}[\![\tau' \,/\, \overline{op_i : \sigma_i \Rightarrow \tau_i}]\!]$ is the least fixed point of the following function.

$$
\begin{aligned}
\mathcal{E}'(X) = {} & \{e \mid \exists v \in \mathcal{V}[\![\tau']\!].\, e \to^* v\} \\
& \cup \{e \\
& \quad \mid \exists K, i, v.\, e \to^* K[op_i\ v] \\
& \quad \wedge K\ op_i\text{-free} \\
& \quad \wedge v \in \mathcal{V}[\![\sigma_i]\!] \\
& \quad \wedge \forall u \in \mathcal{V}[\![\tau_i]\!].\, K[u] \in X\}
\end{aligned}
$$

The function $\mathcal{E}'$ is monotone, so it has a least fixed point by the Knaster-Tarski theorem.

Examples (blackboard)

$$\mathrm{ask}()\ \mathsf{xor}\ \mathrm{ask}() : \mathsf{bool}\ /\ \mathrm{ask} : \mathsf{unit} \Rightarrow \mathsf{bool}$$

Closing

$$\mathcal{G}[\![\Gamma]\!] = \{\gamma : \mathrm{Var} \to \mathrm{Val} \mid \forall x : \tau \in \Gamma.\, \gamma(x) \in \mathcal{V}[\![\tau]\!]\}$$

$$\Gamma \models e : \tau \;/\; \rho \iff \forall \gamma \in \mathcal{G}[\![\Gamma]\!].\, \gamma(e) \in \mathcal{E}[\![\tau/\rho]\!]$$

Simple typing

$$\tau ::= b \mid \tau \rightarrow_\rho \tau \qquad \varepsilon ::= op : \tau \Rightarrow \tau \qquad \rho ::= \cdot \mid \varepsilon \cdot \rho$$

$$\frac{x : \tau \in \Gamma}{\Gamma \vdash x : \tau \;/\; \cdot} \qquad \frac{\Gamma, x : \tau_1 \vdash e : \tau_2 \;/\; \rho}{\Gamma \vdash \lambda x.\, e : \tau_1 \rightarrow_\rho \tau_2 \;/\; \cdot}$$

$$\frac{\Gamma \vdash e_1 : \tau_1 \rightarrow_\rho \tau_2 \;/\; \rho \qquad \Gamma \vdash e_2 : \tau_1 \;/\; \rho}{\Gamma \vdash e_1\, e_2 : \tau_2 \;/\; \rho}$$

$$\frac{\Gamma \vdash v : \tau_1 \;/\; \cdot}{\Gamma \vdash op\ v : \tau_2 \;/\; op : \tau_1 \Rightarrow \tau_2}$$

$$\frac{\Gamma \vdash e : \tau \;/\; op : \tau_1 \Rightarrow \tau_2 \cdot \rho \qquad \Gamma, x : \tau_1, k : \tau_2 \rightarrow_\rho \tau \vdash e_h : \tau \;/\; \rho}{\Gamma \vdash \mathsf{handle}\ e\ \{op\ x\ k \rightarrow e_h\} : \tau \;/\; \rho}$$

Compatibility

For a typing rule

$$\frac{\Gamma, x : \tau_1 \vdash e : \tau_2 \mathbin{/} \rho}{\Gamma \vdash \lambda x.\, e : \tau_1 \rightarrow_\rho \tau_2 \mathbin{/} \cdot}$$

the corresponding compatibility lemma is

Lemma (Lambda compatibility)

If

$$\Gamma, x : \tau_1 \models e : \tau_2 \mathbin{/} \rho$$

then

$$\Gamma \models \lambda x.\, e : \tau_1 \rightarrow_\rho \tau_2 \mathbin{/} \cdot$$

Tying everything together: fundamental theorem

By the compatibility lemmas for each typing rule, we have

$$\Gamma \vdash e : \tau \mathbin{/} \rho \implies \Gamma \models e : \tau \mathbin{/} \rho.$$

In particular, if $\vdash e : \tau \mathbin{/} \cdot$, then $e \in \mathcal{E}[\![\tau / \cdot]\!]$. For the empty row $\cdot$, the only possibility is termination to a value.

Polymorphic typing: kinding

$$\tau ::= \alpha \mid \forall\alpha.\,\tau \mid b \mid \tau \rightarrow_\rho \tau \mid \cdot \mid (op : \tau \Rightarrow \tau) \cdot \rho$$

$$\kappa ::= \mathsf{T} \mid \mathsf{R}$$

$$\frac{\Delta \vdash \tau_1 :: \mathsf{T} \qquad \Delta \vdash \rho :: \mathsf{R} \qquad \Delta \vdash \tau_2 :: \mathsf{T}}{\Delta \vdash \tau_1 \rightarrow_\rho \tau_2 :: \mathsf{T}}$$

$$\frac{\alpha :: \kappa \in \Delta}{\Delta \vdash \alpha :: \kappa} \qquad \frac{\Delta, \alpha :: \kappa \vdash \tau :: \mathsf{T}}{\Delta \vdash \forall\alpha :: \kappa.\,\tau :: \mathsf{T}}$$

$$\frac{}{\Delta \vdash \cdot :: \mathsf{R}} \qquad \frac{\Delta \vdash \tau_1 :: \mathsf{T} \qquad \Delta \vdash \tau_2 :: \mathsf{T} \qquad \Delta \vdash \rho :: \mathsf{R}}{\Delta \vdash (op : \tau_1 \Rightarrow \tau_2) \cdot \rho :: \mathsf{R}}$$

Polymorphic typing

$$\tau ::= \alpha \mid \forall\alpha.\, \tau \mid b \mid \tau \to_\rho \tau \mid \cdot \mid (op : \tau \Rightarrow \tau) \cdot \rho$$

Old rules get a type variable environment Δ, e.g.

$$\frac{\Delta; \Gamma, x : \tau_1 \vdash e : \tau_2 \;/\; \rho}{\Delta; \Gamma \vdash \lambda x.\, e : \tau_1 \to_\rho \tau_2 \;/\; \cdot}$$

Rules pertaining to polymorphism actually use it:

$$\frac{\Delta, \alpha :: \kappa; \Gamma \vdash e : \tau \;/\; \cdot}{\Delta; \Gamma \vdash e : \forall\alpha :: \kappa.\, \tau \;/\; \cdot}$$

$$\frac{\Delta \vdash \sigma :: \kappa \qquad \Delta; \Gamma \vdash e : \forall\alpha :: \kappa.\, \tau \;/\; \rho}{\Delta; \Gamma \vdash e : \tau[\sigma/\alpha] \;/\; \rho}$$

Lift

What if we have a row

$$(op : \sigma \Rightarrow \tau) \cdot \rho$$

and then substitute something that also has op for ρ?

To be able to use the second occurrence of op, we introduce *lift*, which makes operations inside skip the nearest handler:

$$\text{handle handle } (op\ 1) + (\text{lift}^{op}(op\ \text{"}\texttt{text}\text{"}))\,\{op...\}\,\{op...\}$$

Lift

What if we have a row

$$(op : \sigma \Rightarrow \tau) \cdot \rho$$

and then substitute something that also has op for ρ?

To be able to use the second occurrence of op, we introduce *lift*, which makes operations inside skip the nearest handler:

$$\text{handle}\ \text{handle}\ (op\ 1) + (\text{lift}^{op}(op\ \text{"}\texttt{text}\text{"}))\ \{op...\}\ \{op...\}$$

Lift

What if we have a row

$$(op : \sigma \Rightarrow \tau) \cdot \rho$$

and then substitute something that also has op for ρ?

To be able to use the second occurrence of op, we introduce *lift*, which makes operations inside skip the nearest handler:

$$\text{handle}\ \text{handle}\ (op\ 1) + (\text{lift}^{op}(op\ \text{"}\texttt{text}\text{"}))\ \{op...\}\ \{op...\}$$

Lift

What if we have a row

$$(op : \sigma \Rightarrow \tau) \cdot \rho$$

and then substitute something that also has op for ρ?

To be able to use the second occurrence of op, we introduce *lift*, which makes operations inside skip the nearest handler:

$$\text{handle}\ \text{handle}\ (op\ 1) + (\text{lift}^{op}(op\ \text{"}\texttt{text}\text{"}))\ \{op...\}\ \{op...\}$$

Binary relations: interpretation of kinds

$$
\begin{aligned}
[\![\mathsf{T}]\!] &= \mathcal{P}(\mathrm{Val}^2) \\
[\![\mathsf{R}]\!] &= \mathcal{P}(\mathrm{Exp}^2 \times (\mathrm{Op} \to \mathbb{N})^2 \times [\![\mathsf{T}]\!])
\end{aligned}
$$

Binary relations: values and effects

$$
\begin{aligned}
[\![\mathsf{bool}]\!]_\eta &= \{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\} \\
[\![\sigma \to_\rho \tau]\!]_\eta &= \{(\lambda x.\, e_1, \lambda x.\, e_2) \\
&\qquad \mid \forall (v_1, v_2) \in [\![\sigma]\!]_\eta.\, (e_1[v_1/x], e_2[v_2/x]) \in \mathcal{E}[\![\tau/\rho]\!]_\eta\}
\end{aligned}
$$

$$
\begin{aligned}
[\![op : \sigma \Rightarrow \tau]\!]_\eta &= \{(op\ v_1, op\ v_2, (op \mapsto 0), (op \mapsto 0), [\![\tau]\!]_\eta) \\
&\qquad \mid (v_1, v_2) \in [\![\sigma]\!]_\eta\} \\
[\![\cdot]\!]_\eta &= \emptyset \\
[\![(op : \varepsilon) \cdot \rho]\!]_\eta &= [\![op : \varepsilon]\!]_\eta \cup ([\![\rho]\!]_\eta \uparrow^{op})
\end{aligned}
$$

$$
\begin{aligned}
[\![\alpha]\!]_\eta &= \eta(\alpha) \\
[\![\forall \alpha :: \kappa.\, \tau]\!]_\eta &= \bigcap_{\mu \in [\![\kappa]\!]} [\![\tau]\!]_{\eta[\alpha \mapsto \mu]}
\end{aligned}
$$

Binary relations: expressions

$$
\begin{aligned}
\mathcal{E}[\![\tau \mathbin{/} \rho]\!]_\eta = {} & \{(e_1, e_2) \mid \exists (v_1, v_2) \in [\![\tau]\!]_\eta .\, e_i \to^* v_i\} \\
& \cup \{(e_1, e_2) \\
& \quad \mid \exists K_1, K_2, (e'_1, e'_2, f_1, f_2, \mu) \in [\![\rho]\!]_\eta .\, e_i \to^* K_i[e'_i] \\
& \quad \wedge K_i\ f_i\text{-free} \\
& \quad \wedge \forall (u_1, u_2) \in \mu, (K_1[u_1], K_2[u_2]) \in \mathcal{E}[\![\tau \mathbin{/} \rho]\!]_\eta\}
\end{aligned}
$$

Equivalence examples (blackboard)

$$f : \forall\alpha.\,(1 \to_\alpha \mathsf{int}) \to_\alpha \tau \models$$
$$\mathsf{handle}\ f\ \mathrm{ask}\ \{\mathrm{ask}\ \_\ k \to k\ 5\} \approx f\ (\lambda x.\,5) : \tau \;/\; \cdot$$

Problem 1: Finish the equivalence example

We have

$$R = \Big\{ \big(\mathrm{ask}\ (), (\lambda x.\, 5)\ (), (\mathrm{ask} \mapsto 0), \emptyset, \{(5,5)\} \big) \Big\} \tag{1}$$

$$(f_1\ \mathrm{ask}, f_2\ (\lambda x.\, 5)) \in \mathcal{E}[\![\tau \ /\ \alpha]\!]_{[\alpha \mapsto R]} \tag{2}$$

Show by induction on (2) that

$$(\text{handle } f_1\ \mathrm{ask}\ \{\mathrm{ask}\ x\ k \to k\ 5\}, f_2\ (\lambda x.\, 5)) \in \mathcal{E}[\![\tau \ /\ \cdot]\!]_{\emptyset}$$

You should only need to know this about freeness:
$K\ (\mathrm{ask} \mapsto 0)\text{-free}$ means handle $K[\mathrm{ask}()]\{\mathrm{ask}...\}$ can be reduced.

Problem 2: Interpretation of polymorphic operations

Consider polymorphic operations, e.g.

$$(\mathrm{id}\ (\lambda x.\, x + 2))\ (\mathrm{id}\ 40) : \mathsf{int}\ /\ \mathrm{id} : \alpha{::}\mathsf{T}.\, \alpha \Rightarrow \alpha$$

(This is a different feature than polymorphic effects that let you e.g. have a state int handler and state string handler.)
Which semantic interpretation is correct? Why?

$$[\![\mathrm{op} : \alpha{::}\kappa.\, \sigma \Rightarrow \tau]\!]_\eta = \bigcap_{\mu \in [\![\kappa]\!]} [\![\mathrm{op} : \sigma \Rightarrow \tau]\!]_{\eta[\alpha \mapsto \mu]} \tag{3}$$

$$[\![\mathrm{op} : \alpha{::}\kappa.\, \sigma \Rightarrow \tau]\!]_\eta = \bigcup_{\mu \in [\![\kappa]\!]} [\![\mathrm{op} : \sigma \Rightarrow \tau]\!]_{\eta[\alpha \mapsto \mu]} \tag{4}$$