Polymorphic algebraic effects: theoretical properties and implementation

Wiktor Kuchta

Chapter 1

Introduction

A programming language needs to be more than just a lambda calculus, capable only of functional abstraction and evaluation of expressions. Programs need to have an effect on the outside world and, thinking more locally, in our programs we would like to have fragments which do not merely reduce to a value in isolation, but affect the execution of surrounding code in interesting ways. This is what we call computational effects, the typical examples of which are: input/output, mutable state, exceptions, nondeterminism, and coroutines.

Today, we are at the mercy of programming language designers to include the effects we would like to use and make sure that they all interact with each other well. In recent years, however, a new approach, called *algebraic effects*, has emerged. Algebraic effects allow programmers to express all the usual computational effects as library code and use them in the usual direct style (as opposed to eg. monadic encodings, which do not always play well with the rest of the language and do not compose easily). These implementations of effects are instances of one concept and hence interact with each other predictably.

More concretely, algebraic effects may be thought of as resumable exceptions. We can *perform* an *operation*, just as we can raise an exception in a typical high-level programming language. The operation is then handled by the nearest enclosing *handler* for the specific operation. The handler receives the value given at perform-point and also, unlike normal exception handlers, a *continuation*, a first-class functional value representing the rest of the code to execute inside the handler from the perform-point. By calling the continuation with a value we can resume at the perform-point, as if performing the operation evaluated to the value. But more interestingly, we can resume multiple times, or never, or store the continuation for later use.

This access to the continuation, while powerful, might also be a recipe for disaster. That is why languages with algebraic effects typically have *type-and-effect systems*, not only tracking the type of value an expression might evaluate to, but also what kind of effects it may perform on the way.

This work will study a particular formal language equipped with algebraic effects. Although most results in the domain of programming languages depend crucially on the exact calculus and type system used, we hope that the results, ideas, and techniques may apply elsewhere. This should not seem far-fetched, as the paper which introduced the language [2] is exactly about interexpressibility with other systems.

[todo other contributions]

I would like to thank Dariusz Biernacki, Filip Sieczkowski, and Piotr Polesiuk for their help with this work.

Chapter 2

The language

We will study the deep handler calculus and type-and-effect system formulated in [2]. It is a refreshingly minimal language—the call-by-value lambda calculus with a few extensions to be able to express the essence of algebraic effects. There is only one unnamed universal operation, performed $\mathsf{do}\, v$. To be able to simulate calculi with named effects (and more), the *lift* operator, written $[e]$, is introduced. When operations are performed inside the expression e enclosed by lift, the nearest handler will be skipped and the operation will be handled by the next one instead. Naturally, the operator composes, so multiple enclosing lifts means multiple handlers skipped. In contrast to most work on algebraic effects, the effect-tracking system here is structural, we do not have any concept of predefined or user-defined (named) signatures of effects. Finally, the language features polymorphic expressions and polymorphic operations.

$$
\begin{aligned}
\mathrm{TVar} &\ni \alpha, \beta, \ldots \\
\mathrm{Var} &\ni f, r, x, y, \ldots \\
\mathrm{Kind} \ni \kappa &::= \mathsf{T} \mid \mathsf{E} \mid \mathsf{R} \\
\mathrm{Typelike} \ni \sigma, \tau, \varepsilon, \rho &::= \alpha \mid \tau \to_\rho \tau \mid \forall \alpha :: \kappa.\, \tau \mid \iota \mid \tau \Rightarrow \tau \mid \varepsilon \cdot \rho \\
\mathrm{Val} \ni v, u &::= x \mid \lambda x.\, e \\
\mathrm{Exp} \ni e &::= v \mid e\ e \mid [e] \mid \mathsf{do}\ v \mid \mathsf{handle}\ e\,\{x, r.\, e;\ x.\, e\} \\
\mathrm{ECont} \ni K &::= \square \mid K\ e \mid v\ K \mid [K] \mid \mathsf{handle}\ K\,\{x, r.\, e;\ x.\, e\}
\end{aligned}
$$

Figure 2.1: Syntax.

$$\frac{}{0\text{-free}(\square)} \qquad \frac{n\text{-free}(K)}{n\text{-free}(K\ e)} \qquad \frac{n\text{-free}(K)}{n\text{-free}(v\ K)} \qquad \frac{n\text{-free}(K)}{n+1\text{-free}([K])}$$

$$\frac{n+1\text{-free}(K)}{n\text{-free}(\mathsf{handle}\ K\ \{x, r.\, e_h;\ x.\, e_r\})}$$

Figure 2.2: Evaluation context freeness.

$$\frac{e_1 \mapsto e_2}{K[e_1] \to K[e_2]} \qquad (\lambda x.\, e)\ v \mapsto e\{v/x\} \qquad [v] \mapsto v$$

$$\frac{0\text{-free}(K) \qquad v_c = \lambda z.\, \mathsf{handle}\ K[z]\ \{x, r.\, e_h;\ x.\, e_r\}}{\mathsf{handle}\ K[\mathsf{do}\ v]\ \{x, r.\, e_h;\ x.\, e_r\} \mapsto e_h\{v/x\}\{v_c/r\}}$$

$$\mathsf{handle}\ v\ \{x, r.\, e_h;\ x.\, e_r\} \mapsto e_r\{v/x\}$$

Figure 2.3: Single-step reduction.

$$\frac{\alpha :: \kappa \in \Delta}{\Delta \vdash \alpha :: \kappa} \qquad \frac{\Delta \vdash \tau_1 :: \mathsf{T} \qquad \Delta \vdash \rho :: \mathsf{R} \qquad \Delta \vdash \tau_2 :: \mathsf{T}}{\Delta \vdash \tau_1 \to_\rho \tau_2 :: \mathsf{T}} \qquad \frac{\Delta, \alpha :: \kappa \vdash \tau :: \mathsf{T}}{\Delta \vdash \forall \alpha :: \kappa.\, \tau :: \mathsf{T}}$$

$$\frac{}{\Delta \vdash \iota :: \mathsf{R}} \qquad \frac{\Delta \vdash \varepsilon :: \mathsf{E} \qquad \Delta \vdash \rho :: \mathsf{R}}{\Delta \vdash \varepsilon \cdot \rho :: \mathsf{R}} \qquad \frac{\Delta, \Delta' \vdash \tau_1 :: \mathsf{T} \qquad \Delta, \Delta' \vdash \tau_2 :: \mathsf{T}}{\Delta \vdash \Delta'.\, \tau_1 \Rightarrow \tau_2 :: \mathsf{E}}$$

$$\Delta \vdash \delta :: \Delta' \iff \mathrm{dom}(\delta) = \mathrm{dom}(\Delta') \wedge \forall \alpha \in \mathrm{dom}(\delta).\, \Delta \vdash \delta(\alpha) :: \Delta'(\alpha)$$

Figure 2.4: Well-formedness of types, rows, and type substitution.

$$\frac{}{\Delta \vdash \sigma <: \sigma} \qquad \frac{\Delta \vdash \tau_2^1 <: \tau_1^1 \qquad \Delta \vdash \rho_1 <: \rho_2 \qquad \Delta \vdash \tau_1^2 <: \tau_2^2}{\Delta \vdash \tau_1^1 \to_{\rho_1} \tau_1^2 <: \tau_2^1 \to_{\rho_2} \tau_2^2}$$

$$\frac{\Delta, \alpha :: \kappa \vdash \tau_1 <: \tau_2}{\Delta \vdash \forall \alpha :: \kappa.\, \tau_1 <: \forall \alpha :: \kappa.\, \tau_2} \qquad \frac{\Delta \vdash \rho :: \mathsf{R}}{\Delta \vdash \iota <: \rho} \qquad \frac{\Delta \vdash \rho_1 <: \rho_2}{\Delta \vdash \varepsilon \cdot \rho_1 <: \varepsilon \cdot \rho_2}$$

Figure 2.5: Subtyping.

$$\frac{x : \tau \in \Gamma}{\Delta; \Gamma \vdash x : \tau \mathrel{/} \iota} \qquad \frac{\Delta \vdash \tau_1 :: \mathsf{T} \qquad \Delta; \Gamma, x : \tau_1 \vdash e : \tau_2 \mathrel{/} \rho}{\Delta; \Gamma \vdash \lambda x.\, e : \tau_1 \to_\rho \tau_2 \mathrel{/} \iota}$$

$$\frac{\Delta; \Gamma \vdash e_1 : \tau_1 \to_\rho \tau_2 \mathrel{/} \rho \qquad \Delta; \Gamma \vdash e_2 : \tau_1 \mathrel{/} \rho}{\Delta; \Gamma \vdash e_1\, e_2 : \tau_2 \mathrel{/} \rho} \qquad \frac{\Delta \vdash \varepsilon :: \mathsf{E} \qquad \Delta; \Gamma \vdash e : \tau \mathrel{/} \rho}{\Delta; \Gamma \vdash [e] : \tau \mathrel{/} \varepsilon \cdot \rho}$$

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

$$\frac{\Delta \vdash \tau_1 <: \tau_2 \qquad \Delta \vdash \rho_1 <: \rho_2 \qquad \Delta; \Gamma \vdash e : \tau_1 \mathrel{/} \rho_1}{\Delta; \Gamma \vdash e : \tau_2 \mathrel{/} \rho_2}$$

$$\frac{\Delta; \Gamma \vdash v : \delta(\tau_1) \mathrel{/} \iota \qquad \Delta \vdash \delta :: \Delta' \qquad \Delta \vdash \Delta'.\, \tau_1 \Rightarrow \tau_2 :: \mathsf{E}}{\Delta; \Gamma \vdash \mathsf{do}\ v : \delta(\tau_2) \mathrel{/} (\Delta'.\, \tau_1 \Rightarrow \tau_2)}$$

$$\frac{\begin{array}{c}\Delta; \Gamma \vdash e : \tau \mathrel{/} (\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho \\ \Delta, \Delta'; \Gamma, x : \tau_1, r : \tau_2 \to_\rho \tau_r \vdash e_h : \tau_r \mathrel{/} \rho \qquad \Delta; \Gamma, x : \tau \vdash e_r : \tau_r \mathrel{/} \rho\end{array}}{\Delta; \Gamma \vdash \mathsf{handle}\ e\ \{x, r.\, e_h; x.\, e_r\} : \tau_r \mathrel{/} \rho}$$

Figure 2.6: Type system.

Chapter 3

The logical relation

[todo introduction to logical relations]

The logical relation is adapted from [1] with a few changes. Some changes are due to language differences: we have only one universal operation which simplifies the treatment of effects, polymorphism does not manifest at the expression level—we do not have type lambdas, and our operations can be polymorphic. Instead of a binary step-indexed relation, our goal is to build a unary relation without step-indexing.

3.1 Definition

First, we define the spaces of *semantic types* and *semantic effects*, which are also the interpretations of the appropriate kinds:

$$\begin{aligned} [\![\mathsf{T}]\!] &= \mathcal{P}(\mathrm{Val}) = \mathsf{Type}, \\ [\![\mathsf{E}]\!] = [\![\mathsf{R}]\!] &= \mathcal{P}(\mathrm{Exp} \times \mathbb{N} \times \mathsf{Type}) = \mathsf{Eff}. \end{aligned}$$

Semantic types are simply sets of closed values. Semantic effects are sets of triples, which aim to describe a situation in which an expression being evaluated performs an effect. The components of such a triple are: the argument of the operation, the freeness of the enclosing context beyond which the operation can be handled, and the semantic type of values we can call the resumption with.

Logical relations can be used to establish various properties. Here, we choose termination to a value as our *observation*:

$$\mathrm{Obs} = \{e \mid \exists v.\, e \to^* v\}.$$

We start with defining interpretations of types and effects. We parameterize the definitions by a mapping η from type variables to semantic types or effects.

$$\begin{aligned} [\![\tau_1 \to_\rho \tau_2]\!]_\eta &= \{\lambda x.\, e \mid \forall v \in [\![\tau_1]\!]_\eta.\, e\{v/x\} \in \mathcal{E}[\![\tau_2/\rho]\!]_\eta\} \\ [\![\forall \alpha :: \kappa.\, \tau]\!]_\eta &= \{v \mid \forall \mu \in [\![\kappa]\!].\, v \in [\![\tau]\!]_{[\alpha \mapsto \mu]\eta}\} \\ [\![\alpha]\!]_\eta &= \{v \mid \forall K.\, (\forall u \in \eta(\alpha).\, K[u] \in \mathrm{Obs}) \implies K[v] \in \mathrm{Obs}\} \end{aligned}$$

$$
\begin{aligned}
[\![\tau_1 \Rightarrow \tau_2]\!]_\eta &= \{(v, 0, [\![\tau_2]\!]_\eta) \mid v \in [\![\tau_1]\!]_\eta\} \\
[\![\forall\alpha :: \kappa.\, \varepsilon]\!]_\eta &= \{t \mid \exists \mu \in [\![\kappa]\!].\, t \in [\![\varepsilon]\!]_{[\alpha \mapsto \mu]\eta}\} \\
[\![\varepsilon \cdot \rho]\!]_\eta &= [\![\varepsilon]\!]_\eta \cup \{(v, n+1, \mu) \mid (v, n, \mu) \in [\![\rho]\!]_\eta\} \\
[\![\iota]\!]_\eta &= \emptyset
\end{aligned}
$$

By adapting the definitions from [1] we end up with the following equations we wish to solve, defining the sets of good expressions, contexts, and control-stuck terms:

$$
\begin{aligned}
\mathcal{E}[\![\tau/\rho]\!]_\eta &= \{e \mid \forall K \in \mathcal{K}[\![\tau/\rho]\!]_\eta.\, K[e] \in \mathrm{Obs}\} \\
\mathcal{K}[\![\tau/\rho]\!]_\eta &= \{K \mid \forall v \in [\![\tau]\!]_\eta.\, K[v] \in \mathrm{Obs} \land \forall s \in \mathcal{S}[\![\tau/\rho]\!]_\eta.\, K[s] \in \mathrm{Obs}\} \\
\mathcal{S}[\![\tau/\rho]\!]_\eta &= \{K[\mathsf{do}\ v] \mid \exists n, \mu.\, (v, n, \mu) \in [\![\rho]\!]_\eta \land n\text{-free}(K) \land \forall u \in \mu.\, K[u] \in \mathcal{E}[\![\tau/\rho]\!]_\eta\}
\end{aligned}
$$

We untangle the recursion firstly by abstracting out the recursive occurence as an additional parameter and secondly by noticing that we do not need the exact interpretations of types and effects, any semantic type and effect will do.

$$
\begin{aligned}
\mathcal{E}(\mu, \xi)(X) &= \{e \mid \forall K \in \mathcal{K}(\mu, \xi)(X).\, K[e] \in \mathrm{Obs}\} \\
\mathcal{K}(\mu, \xi)(X) &= \{K \mid \forall v \in \mu.\, K[v] \in \mathrm{Obs} \land \forall s \in \mathcal{S}(\mu, \xi)(X).\, K[s] \in \mathrm{Obs}\} \\
\mathcal{S}(\_, \xi)(X) &= \{K[\mathsf{do}\ v] \mid \exists n, \mu.\, (v, n, \mu) \in \xi \land n\text{-free}(K) \land \forall u \in \mu.\, K[u] \in X\}
\end{aligned}
$$

Lemma 1. *If $X_1 \subseteq X_2$ then $\mathcal{S}(\_, \xi)(X_1) \subseteq \mathcal{S}(\_, \xi)(X_2)$, $\mathcal{K}(\_, \xi)(X_1) \supseteq \mathcal{K}(\_, \xi)(X_2)$, and $\mathcal{E}(\_, \xi)(X_1) \subseteq \mathcal{E}(\_, \xi)(X_2)$.*

We note that $\mathcal{E}(\mu, \xi)$ is non-decreasing, i.e. it is a monotone function on the complete lattice of the powerset of Exp ordered by inclusion. Hence, by the Knaster-Tarski theorem, it has a greatest fixed point, which we denote $\nu\mathcal{E}(\mu, \xi)$. Moreover, the theorem states that it is specified by the following join:

$$
\nu\mathcal{E}(\mu, \xi) = \bigcup \{X \subseteq \mathrm{Exp} \mid X \subseteq \mathcal{E}(\mu, \xi)(X)\}.
$$

Now, it remains to plug in the interpretations of types as the semantic types:

$$
\begin{aligned}
\mathcal{E}[\![\tau/\rho]\!]_\eta &= \nu\mathcal{E}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta), \\
\mathcal{K}[\![\tau/\rho]\!]_\eta &= \mathcal{K}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta)(\mathcal{E}[\![\tau/\rho]\!]_\eta), \\
\mathcal{S}[\![\tau/\rho]\!]_\eta &= \mathcal{S}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta)(\mathcal{E}[\![\tau/\rho]\!]_\eta).
\end{aligned}
$$

Only now we can finally say that the interpretations of types and effects are defined by structural induction.

3.2 Basic properties

Lemma 2 (Coinduction principle)**.** *If $X \subseteq \mathcal{E}(\mu, \xi)(X)$ then $X \subseteq \nu\mathcal{E}(\mu, \xi)$.*

Proof. Follows directly from the definition. □

Lemma 3 (Strong coinduction). *If $X \subseteq \mathcal{E}(\mu,\xi)(X \cup \nu\mathcal{E}(\mu,\xi))$, then $X \subseteq \nu\mathcal{E}(\mu,\xi)$.*

Proof. We have $\nu\mathcal{E}(\mu,\xi) = \mathcal{E}(\mu,\xi)(\nu\mathcal{E}(\mu,\xi)) \subseteq \mathcal{E}(\mu,\xi)(X \cup \nu\mathcal{E}(\mu,\xi))$. Combining with the lemma assumption we obtain $X \cup \nu\mathcal{E}(\mu,\xi) \subseteq \mathcal{E}(\mu,\xi)(X \cup \nu\mathcal{E}(\mu,\xi))$, which is what we want by lemma 2. □

To reduce notational burden in proofs by strong coinduction, we introduce the shorthand $\mathcal{E}[\![\tau/\rho]\!]_\eta^{\cup Y}$ to mean $\mathcal{E}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta)(Y \cup \mathcal{E}[\![\tau/\rho]\!]_\eta)$. Similarly, $\mathcal{K}[\![\tau/\rho]\!]_\eta^{\cup Y}$ means $\mathcal{K}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta)(Y \cup \mathcal{E}[\![\tau/\rho]\!]_\eta)$ and $\mathcal{S}[\![\tau/\rho]\!]_\eta^{\cup Y}$ means $\mathcal{S}([\![\tau]\!]_\eta, [\![\rho]\!]_\eta)(Y \cup \mathcal{E}[\![\tau/\rho]\!]_\eta)$.

Lemma 4. *If $\mu_1 \subseteq \mu_2$ and $\xi_1 \subseteq \xi_2$, then*

$$\begin{aligned}
\mathcal{S}(\mu_1,\xi_1)(X) &\subseteq \mathcal{S}(\mu_2,\xi_2)(X),\\
\mathcal{K}(\mu_1,\xi_1)(X) &\supseteq \mathcal{K}(\mu_2,\xi_2)(X),\\
\mathcal{E}(\mu_1,\xi_1)(X) &\subseteq \mathcal{E}(\mu_2,\xi_2)(X).
\end{aligned}$$

Lemma 5. *If $\mu_1 \subseteq \mu_2$ and $\xi_1 \subseteq \xi_2$, then $\nu\mathcal{E}(\mu_1,\xi_1) \subseteq \nu\mathcal{E}(\mu_2,\xi_2)$.*

Proof. By lemma 5 and transitivity if $\mu_1 \subseteq \mu_2$ and $\xi_1 \subseteq \xi_2$, then $X \subseteq \mathcal{E}(\mu_1,\xi_1)(X)$ implies $X \subseteq \mathcal{E}(\mu_2,\xi_2)(X)$. Therefore

$$\{X \subseteq \mathrm{Exp} \mid X \subseteq \mathcal{E}(\mu_1,\xi_1)(X)\} \subseteq \{X \subseteq \mathrm{Exp} \mid X \subseteq \mathcal{E}(\mu_2,\xi_2)(X)\},$$

$$\bigcup\{X \subseteq \mathrm{Exp} \mid X \subseteq \mathcal{E}(\mu_1,\xi_1)(X)\} \subseteq \bigcup\{X \subseteq \mathrm{Exp} \mid X \subseteq \mathcal{E}(\mu_2,\xi_2)(X)\}.$$

□

Lemma 6. *For any semantic type μ we have $\square \in \mathcal{K}(\mu,\emptyset)$.*

Proof. The set $\mathcal{S}(\mu,\emptyset)$ is empty and μ is contained in Obs. □

Lemma 7 (Value inclusion). *For any τ and ρ we have $[\![\tau]\!]_\eta \subseteq \mathcal{E}[\![\tau/\rho]\!]_\eta$.*

Proof. Follows from the definition of $\mathcal{E}[\![\tau/\rho]\!]_\eta$ and $\mathcal{K}[\![\tau/\rho]\!]_\eta$. □

Lemma 8 (Control-stuck inclusion). *For any τ and ρ we have $\mathcal{S}[\![\tau/\rho]\!]_\eta \subseteq \mathcal{E}[\![\tau/\rho]\!]_\eta$.*

Proof. Follows from the definition of $\mathcal{E}[\![\tau/\rho]\!]_\eta$ and $\mathcal{K}[\![\tau/\rho]\!]_\eta$. □

3.3 Compatibility lemmas

We want to establish that $\vdash e : \tau \,/\, \iota$ implies $e \in \mathcal{E}[\![\tau/\iota]\!]$. Termination to a value will follow from lemma 6.

For this purpose we will prove a semantic counterpart of each typing rule. First, we need to define a counterpart to the typing judgment. Unlike typing judgments, our relations are on closed terms only, so we get around that by using substitution. We define semantic entailment as

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

where $[\![\Delta]\!] = \{\eta \mid \forall \alpha :: \kappa \in \Delta.\, \eta(\alpha) \in [\![\kappa]\!]\}$ contains type-level mappings and $[\![\Gamma]\!]_\eta = \{\gamma \mid \forall x : \tau \in \Gamma.\, \gamma(x) \in [\![\tau]\!]_\eta\}$ contains expression-level variable substitutions.

Lemma 9 (Variable compatibility)**.**

Proof. Assume $x : \tau \in \Gamma$. We want to prove $\Delta; \Gamma \models x : \tau \,/\, \iota$. Take any $\eta \in [\![\Delta]\!]$ and $\gamma \in [\![\Gamma]\!]_\eta$. We want to show $\gamma(x) \in \mathcal{E}[\![\tau/\iota]\!]_\eta$. From the definition of $[\![\Gamma]\!]_\eta$ we know that $\gamma(x) \in [\![\tau]\!]_\eta$, so by lemma 7 we have $\gamma(x) \in \mathcal{E}[\![\tau/\iota]\!]_\eta$. □

Lemma 10 (Abstraction compatibility)**.**

Proof. Assume $\Delta \vdash \tau_1 :: \mathsf{T}$ and $\Delta; \Gamma, x : \tau_1 \models e : \tau_2 \,/\, \rho$. We want to prove $\Delta; \Gamma \models \lambda x.\, e : \tau_1 \to_\rho \tau_2 \,/\, \iota$. Take any $\eta \in [\![\Delta]\!]$ and $\gamma \in [\![\Gamma]\!]_\eta$. By lemma 7 it suffices to show $\gamma(\lambda x.\, e) = \lambda x.\, \gamma(e) \in [\![\tau_1 \to_\rho \tau_2]\!]_\eta$. So take any $v \in [\![\tau_1]\!]_\eta$. We need to show $\gamma(e)\{v/x\} \in \mathcal{E}[\![\tau_2/\rho]\!]_\eta$. Let $\gamma' = \gamma'[x \mapsto v]$. Then $\gamma' \in [\![\Gamma, x : \tau_1]\!]_\eta$, so $\gamma(e)\{v/x\} = \gamma'(e) \in \mathcal{E}[\![\tau_2/\rho]\!]_\eta$. □

Lemma 11 (Application compatibility)**.**

Proof. Fix any well-formed Δ and $\tau_1 \to_\rho \tau_2$. Take any $\eta \in [\![\Delta]\!]$. Let $Y = \{e_1\ e_2 \mid e_1 \in \mathcal{E}[\![\tau_1 \to_\rho \tau_2/\rho]\!]_\eta, e_2 \in \mathcal{E}[\![\tau_1/\rho]\!]_\eta\}$. We want to show $Y \subseteq \mathcal{E}[\![\tau_2/\rho]\!]_\eta$. We will proceed by strong coinduction.

We will show $Y \subseteq \mathcal{E}[\![\tau_2/\rho]\!]_\eta^{\cup Y}$. Take any $e_1\ e_2 \in Y$ and $K \in \mathcal{K}[\![\tau_2/\rho]\!]_\eta^{\cup Y}$. We want to show $K[e_1\ e_2] \in \mathrm{Obs}$. Let $K_2 = K[\square\ e_2]$. It suffices to show $K_2 \in \mathcal{K}[\![\tau_1 \to_\rho \tau_2/\rho]\!]_\eta$.

Take any $K'[\mathsf{do}\ v] \in \mathcal{S}[\![\tau_1 \to_\rho \tau_2/\rho]\!]_\eta$. There exists $(v, n, \mu) \in [\![\rho]\!]_\eta$ such that n-free(K') and for all $u \in \mu$ we have $K'[u] \in \mathcal{E}[\![\tau_1 \to_\rho \tau_2/\rho]\!]_\eta$. We want $K[K'[\mathsf{do}\ v]\ e_2] \in \mathrm{Obs}$. It suffices to show $K'[\mathsf{do}\ v]\ e_2 \in \mathcal{S}[\![\tau_2/\rho]\!]_\eta^{\cup Y}$. Which holds, since n-free$(K'\ e_2)$ and for all $u \in \mu$ we have $K'[u]\ e_2 \in Y$.

Now take any $v = (\lambda x.\, e) \in [\![\tau_1 \to_\rho \tau_2]\!]_\eta$. By lemma 7, $v \in \mathcal{E}[\![\tau_1 \to_\rho \tau_2/\rho]\!]_\eta$. We want $K_2[v] = K[v\ e_2] \in \mathrm{Obs}$. It suffices to show that $K_v = K[v\ \square] \in \mathcal{K}[\![\tau_1/\rho]\!]_\eta$.

Take any $K'[\mathsf{do}\ v'] \in \mathcal{S}[\![\tau_1/\rho]\!]_\eta$. There exists $(v', n, \mu) \in [\![\rho]\!]_\eta$ such that n-free(K') and for all $u \in \mu$ we have $K'[u] \in \mathcal{E}[\![\tau_1/\rho]\!]_\eta$. We want $K[v\ K'[\mathsf{do}\ v']] \in \mathrm{Obs}$. It suffices to show $v\ K'[\mathsf{do}\ v'] \in \mathcal{S}[\![\tau_2/\rho]\!]_\eta^{\cup Y}$. Which holds, since n-free$(v\ K')$ and for all $u \in \mu$ we have $v\ K'[u] \in Y$.

Finally, take any $u \in [\![\tau_1]\!]_\eta$. We want $K[v\ u] \in \mathrm{Obs}$. The expression reduces to $K[e\{u/x\}]$, where $e\{u/x\} \in \mathcal{E}[\![\tau_2/\rho]\!]_\eta$. Since $K \in \mathcal{K}[\![\tau_2/\rho]\!]_\eta^{\cup Y} \subseteq \mathcal{K}[\![\tau_2/\rho]\!]_\eta$, $K[e\{u/x\}] \in \mathrm{Obs}$.

□

Lemma 12 (Lift compatibility)**.**

Proof. Assume $\Delta \vdash \tau :: \mathsf{T}$, $\Delta \vdash \varepsilon :: \mathsf{E}$, and $\Delta \vdash \rho :: \mathsf{R}$. Take any $\eta \in [\![\Delta]\!]$. Let $Y = \{[e] \mid e \in \mathcal{E}[\![\tau/\rho]\!]_\eta\}$. We want to show $Y \subseteq \mathcal{E}[\![\tau/\varepsilon \cdot \rho]\!]_\eta$. We will proceed by strong coinduction.

We will show $Y \subseteq \mathcal{E}[\![\tau/\varepsilon \cdot \rho]\!]_\eta^{\cup Y}$. Take any $[e] \in Y$ and $K \in \mathcal{K}[\![\tau/\varepsilon \cdot \rho]\!]_\eta^{\cup Y}$. We want to show $K[[e]] \in \mathrm{Obs}$. It suffices to show that $K[[\square]] \in \mathcal{K}[\![\tau/\rho]\!]_\eta$.

Take any $K'[\mathsf{do}\ v] \in \mathcal{S}[\![\tau/\rho]\!]_\eta$. There exists $(v, n, \mu) \in [\![\rho]\!]_\eta$ such that n-free(K') and for all $u \in \mu$ we have $K'[u] \in \mathcal{E}[\![\tau/\rho]\!]_\eta$. We want $K[[K'[\mathsf{do}\ v]]] \in \mathrm{Obs}$. It suffices to show that $[K'[\mathsf{do}\ v]] \in \mathcal{S}[\![\tau/\varepsilon \cdot \rho]\!]_\eta^{\cup Y}$. Which holds, since $(v, n+1, \mu) \in [\![\varepsilon \cdot \rho]\!]_\eta$, $n+1$-free$([K'])$, and for all $u \in \mu$ we have $[K'[u]] \in Y$.

Now take any $v \in [\![\tau]\!]_\eta$. By lemma 7, $v \in \mathcal{E}[\![\tau/\varepsilon \cdot \rho]\!]_\eta \subseteq \mathcal{E}[\![\tau/\varepsilon \cdot \rho]\!]_\eta^{\cup Y}$. We want to show $K[[v]] \in \mathrm{Obs}$. The expression reduces to $K[v]$, which is in Obs since $K \in \mathcal{K}[\![\tau/\varepsilon \cdot \rho]\!]_\eta^{\cup Y}$. □

Lemma 13 (Handle compatibility).

Proof. Assume $\Delta, \Delta'; \Gamma, x : \tau_1, r : \tau_2 \to_\rho \tau_r \models e_h : \tau_r \,/\, \rho$, $\Delta; \Gamma, x : \tau \models e_r : \tau_r \,/\, \rho$. $\Delta; \Gamma \models e : \tau \,/\, (\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho$. Note that only τ_1 and τ_2 require Δ' to be in context. Let h stand for $\{x, r.\, e_h;\, x.\, e_r\}$. Take any $\eta \in [\![\Delta]\!]$. Let $Y = \{\mathsf{handle}\, e\, h \mid e \in \mathcal{E}[\![\tau/(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta\}$. We want to show $Y \subseteq \mathcal{E}[\![\tau_r/\rho]\!]_\eta$. We will proceed by strong coinduction.

We will show $Y \subseteq \mathcal{E}[\![\tau_r/\rho]\!]_\eta^{\cup Y}$. Take any $e' = \mathsf{handle}\, e\, h \in Y$ and $K \in \mathcal{K}[\![\tau_r/\rho]\!]_\eta^{\cup Y}$. We want to show $K[e'] \in \mathrm{Obs}$. It suffices to show $K' = K[\mathsf{handle}\, \square\, h] \in \mathcal{K}[\![\tau/(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta$.

Take any $v \in [\![\tau]\!]_\eta$. The expression $K'[v]$ reduces to $K[e_r\{v/x\}]$. Which is in Obs, since $e_r\{v/x\} \in \mathcal{E}[\![\tau_r/\rho]\!]_\eta$, and $K \in \mathcal{K}[\![\tau_r/\rho]\!]_\eta$.

Now take any $K''[\mathsf{do}\, v] \in \mathcal{S}[\![\tau/(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta$. There exist $(v, n, \mu) \in [\![(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta$ such that n-free(K'') and for all $u \in \mu$ we have $K''[u] \in \mathcal{E}[\![\tau/(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta$.

If $n = 0$, then $K'[K''[\mathsf{do}\, v]]$ reduces to $K[e_h\{v/x\}\{v_c/r\}]$, where v_c stands for $\lambda z.\, \mathsf{handle}\, K''[z]\, h$. We want to show that this is in Obs, i.e. the expression plugged into K is in $\mathcal{E}[\![\tau_r/\rho]\!]_\eta^{\cup Y}$. From the interpretation of polymorphic effects, there exists η' extending η such that $(v, n, \mu) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta'}$. Therefore $v \in [\![\tau_1]\!]_{\eta'}$ and $\mu = [\![\tau_2]\!]_{\eta'}$. We only need to show $v_c \in [\![\tau_2 \to_\rho \tau_r]\!]_{\eta'}$. So take any $u \in [\![\tau_2]\!]_{\eta'}$. The application $v_c\, u$ reduces to $\mathsf{handle}\, K''[u]\, h$. We know $K''[u] \in \mathcal{E}[\![\tau/(\Delta'.\, \tau_1 \Rightarrow \tau_2) \cdot \rho]\!]_\eta$. We want $\mathsf{handle}\, K''[u]\, h \in \mathcal{E}[\![\tau_r/\rho]\!]_\eta$.

If $n > 0$, then it suffices to show that $\mathsf{handle}\, K''[\mathsf{do}\, v]\, h \in \mathcal{S}[\![\tau_r/\rho]\!]_\eta^{\cup Y}$. Which holds, since $(v, n-1, \mu) \in [\![\rho]\!]_\eta$, $n-1$-free$(\mathsf{handle}\, K''\, h)$, and also $\mathsf{handle}\, K''[u]\, h \in Y$. $\square$

Lemma 14. *If Δ and Δ' disjoint, $\Delta \vdash \delta :: \Delta'$, $\Delta, \Delta' \vdash \tau :: \kappa$, $\eta \in [\![\Delta]\!]$, η' extends η by mappings $\alpha \mapsto [\![\delta(\alpha)]\!]_\eta$, then $[\![\tau]\!]_{\eta'} = [\![\delta(\tau)]\!]_\eta$.*

Proof. By induction on the kinding rules.

If $\tau = \alpha \in \Delta$, then both sides are equal to $\eta(\alpha)$.

If $\tau = \alpha \in \Delta'$, then equality follows from the definition of η'.

If $\tau = \iota$, then both sides are empty.

If $\tau = \forall \alpha :: \kappa.\, \tau' :: \mathsf{T}$, then $[\![\tau]\!]_{\eta'} = \bigcap\{[\![\tau']\!]_{\eta'[\alpha \mapsto \mu]} \mid \mu \in [\![\kappa]\!]\}$ and $[\![\delta(\tau)]\!]_\eta = \bigcap\{[\![\delta(\tau')]\!]_{\eta[\alpha \mapsto \mu]} \mid \mu \in [\![\kappa]\!]\}$, which are equal by the inductive hypothesis.

If $\tau = \forall \alpha :: \kappa.\, \tau' :: \mathsf{E}$, then as above but with unions instead of intersections.

If $\tau = \tau_1 \Rightarrow \tau_2$, then $[\![\tau]\!]_{\eta'} = \{(v, 0, [\![\tau_2]\!]_{\eta'}) \mid v \in [\![\tau_1]\!]_{\eta'}\}$ and $[\![\delta(\tau)]\!]_\eta = \{(v, 0, [\![\delta(\tau_2)]\!]_\eta) \mid v \in [\![\delta(\tau_1)]\!]_\eta\}$, which are equal by the inductive hypothesis.

If $\tau = \varepsilon \cdot \rho$, then $[\![\tau]\!]_{\eta'} = [\![\varepsilon]\!]_{\eta'} \cup \{(v, n+1, \mu) \mid (v, n, \mu) \in [\![\rho]\!]_{\eta'}\}$ and $[\![\delta(\tau)]\!]_\eta = [\![\delta(\varepsilon)]\!]_\eta \cup \{(v, n+1, \mu) \mid (v, n, \mu) \in [\![\delta(\rho)]\!]_\eta\}$, which are equal by the inductive hypothesis. $\square$

Lemma 15 (Do compatibility).

Proof. Assume $\Delta \vdash \delta :: \Delta'$, $\Delta \vdash \Delta'.\, \tau_1 \to \tau_2 :: \mathsf{E}$. Take any $\eta \in [\![\Delta]\!]$. Assume $v \in [\![\delta(\tau_1)]\!]_\eta$. We want to show $\mathsf{do}\, v \in \mathcal{E}[\![\delta(\tau_2)/(\Delta'.\, \tau_1 \Rightarrow \tau_2)]\!]_\eta$.

By lemma 8 it suffices to show $\mathsf{do}\, v \in \mathcal{S}[\![\delta(\tau_2)/(\Delta'.\, \tau_1 \Rightarrow \tau_2)]\!]_\eta$. By taking the empty context in the definition of $\mathcal{S}$ and lemma 7, it suffices to show $(v, 0, [\![\delta(\tau_2)]\!]_\eta) \in$

$[\![\Delta'. \tau_1 \Rightarrow \tau_2]\!]_\eta$. From the interpretation of polymorphic effects, it would be enough to show $(v, 0, [\![\delta(\tau_2)]\!]_\eta) \in [\![\tau_1 \Rightarrow \tau_2]\!]_{\eta'}$, where η' extends η by mappings $\alpha \mapsto [\![\delta(\alpha)]\!]_\eta$ for all $\alpha \in \Delta'$. By the interpretation of operations, it remains to show $v \in [\![\tau_1]\!]_{\eta'}$ and $[\![\delta(\tau_2)]\!]_\eta = [\![\tau_2]\!]_{\eta'}$. Which follows immediately from lemma 14. $\square$

Bibliography

[1] Dariusz Biernacki, Maciej Piróg, Piotr Polesiuk, and Filip Sieczkowski. Handle with care: relational interpretation of algebraic effects and handlers. *Proc. ACM Program. Lang.*, 2(POPL):8:1–8:30, 2018.

[2] Maciej Piróg, Piotr Polesiuk, and Filip Sieczkowski. Typed equivalence of effect handlers and delimited control. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 30:1–30:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.