Basic Operational Preorders for Algebraic Effects

With a pinch of non-determinism and probabilities

Aliaume Lopez

17 / 12 / 2021

Under the supervision of Sylvain Schmitz and Jean Goubault-Larrecq

université PARIS-SACLAY
école normale supérieure paris–saclay
LMF Laboratoire Méthodes Formelles
IRIF INSTITUT DE RECHERCHE EN INFORMATIQUE FONDAMENTALE
Université de Paris

A deceptively long introduction

Effects

PCF+Effects

PCF Plotkin (1977) using call-by-value with effects, similar to Plotkin and Power (2001).

Concrete implementations **with handlers**

- Eff `https://www.eff-lang.org/` Bauer and Pretnar (2012); Plotkin and Pretnar (2013)
- Haskell implementations (Fused Effects, Polysemy, Eff, ...)

What's in my bag?

Local state, global state, exceptions, non-determinism, random numbers, logging, input/output, continuations.

Many of the effects listed are in fact algebraic (modeled by a Lawvere theory) and therefore share nice properties.

Program equivalence and how to deal with it

- Dal Lago et al. (2017)
- Johann et al. (2010)

A deceptively long introduction

What does it look like concretely?

Simple programming language: Types

Non polymorphic term types

$$\tau ::= \textsf{Nat} \mid \tau \to \tau$$

Polymorphic effect types

Example	Type
Lookup	$\sigma : \alpha^{\mathsf{Nat}} \to \alpha$
Binary choice	$\sigma : \alpha^{\mathsf{n}} \to \alpha$
Update	$\sigma : \mathsf{Nat} \times \alpha^{\mathsf{n}} \to \alpha$
Generic case	$\sigma : \mathsf{Nat} \times \alpha^{\mathsf{Nat}} \to \alpha$

What about polymorphism?

Adding polymorphism for terms

- Done in Johann et al. (2010), not the main technicality
- Very useful in practice Wadler (1989); Pitts (2000): "there are not much functions of type $\forall \alpha . \alpha \to \alpha$".

Terms, $\sigma \in \Sigma$

$$\begin{aligned} \mathsf{M} := \mathsf{x} \mid \lambda \mathsf{x} : \tau.\mathsf{M} \mid \mathsf{M}\mathsf{M} \mid \mathsf{fix}\, \mathsf{M} \mid \mathsf{Z} \mid \mathsf{S}\, \mathsf{M} \\ \mid \mathsf{case}\, \mathsf{M}\, \mathsf{of}\, \mathsf{Z} \Rightarrow \mathsf{M}; \mathsf{S}(\mathsf{x}) \Rightarrow \mathsf{M} \\ \mid \sigma(\mathsf{M}, \ldots, \mathsf{M}) \\ \mid \sigma(\mathsf{M}; \mathsf{M}, \ldots, \mathsf{M}) \end{aligned}$$

Values

$$\mathsf{V} := \lambda \mathsf{x} : \tau.\mathsf{M} \mid \mathsf{Z} \mid \mathsf{S}\, \mathsf{V}$$

A deceptively long introduction

Reasoning about effects

Contextual equivalence

$M \equiv_{ctx} M'$

$$\forall C[-], \forall n, C[M] : \mathrm{Nat}, C[M'] : \mathrm{Nat}, C[M] \Downarrow \underline{n} \iff C[M'] \Downarrow \underline{n}$$

Issues

- Can capture free variables;
- More suitable for proving non-equivalence;
- Taylor made for termination.

Two amidst many alternatives

- Bisimilarity and bisimulations Dal Lago et al. (2017);
- Logical relations Johann et al. (2010).

Why can't we easily reason?

In the presence of effects, function extentionality is not a good way to reason:

$$\lambda x.\mathrm{OR}(1, 2) \quad ? \quad \mathrm{OR}(\lambda x.1, \lambda x.2)$$

Why can't we easily reason?

In the presence of effects, function extentionality is not a good way to reason:

$$\lambda x.\mathsf{OR}(1,2) \not\equiv_{\mathsf{ctx}} \mathsf{OR}(\lambda x.1, \lambda x.2)$$

Why can't we easily reason?

In the presence of effects, function extentionality is not a good way to reason:

$$\lambda x.\mathsf{OR}(1, 2) \not\equiv_{\mathsf{ctx}} \mathsf{OR}(\lambda x.1, \lambda x.2)$$

$$\mathsf{C}[-] = (\lambda f.f0 + f0)[-]$$

The zen way of building contextual preorders

The work of Johann et al. (2010)

Ground relations

Stacks, terms, trees

Reduce a pair $\langle S, M \rangle$ to a tree $|S, M|$ of effects where leaves are values.

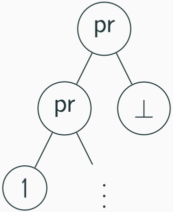
Granted $\preccurlyeq$ is a preorder over $\mathrm{Tree}_{\mathrm{Nat}}$ the free continuous Σ-algebra over Nat.

Contextual Preorder

The contextual preorder is the largest compatible (closed under context) and $\preccurlyeq$-adequate (included in $\preccurlyeq$ at ground type) relation.

Example of trees

Let $\Sigma = \{\mathsf{pr}\}$ be a signature containing one binary effect construction.



Always go concrete

Generic operational meta-theory

Input: A preorder $\preccurlyeq$ for type Nat

Output: A logical relation (!) on programs that characterises contextual preorder (Morris-Style)

Some relations are more equal than others

† Admissible If $t_i \preccurlyeq t'_i$ and $(t_i)_i$, $(t'_i)_i$ are ascending chains then

$$\bigsqcup_i t_i \preccurlyeq \bigsqcup_i t'_i$$

‡ Compatible If $t_i \preccurlyeq t'_i$ and $\sigma \in \Sigma$ then $\sigma(t_1, \ldots) \preccurlyeq \sigma(t'_1, \ldots)$.

◇ Substitutive Given $\rho \colon \mathsf{Tree}_{\mathsf{Nat}} \to \mathsf{Tree}_{\mathsf{Nat}}$, if $t \preccurlyeq t'$ then $t\rho \preccurlyeq t'\rho$

□ Compositional Given $\rho, \rho' \colon \mathsf{Tree}_{\mathsf{Nat}} \to \mathsf{Tree}_{\mathsf{Nat}}$, if $t \preccurlyeq t'$ and $\rho \preccurlyeq \rho'$ then $t\rho \preccurlyeq t'\rho'$

$$\dagger \wedge \ddagger \wedge \diamond \iff \dagger \wedge \Box$$

Examples, counter examples

Effect	Admissible	Compatible	Substitutive	Compositional
Hoare	✓	✓	✓	✓
Smyth	✓	✓	✓	✓
Countable Smyth	✗	?	?	?
Valuations	✓	✓	✓	✓
Exceptions	✓	✓	✓	✓
Mixed Pr/Nd	✓	✓	✗	✗

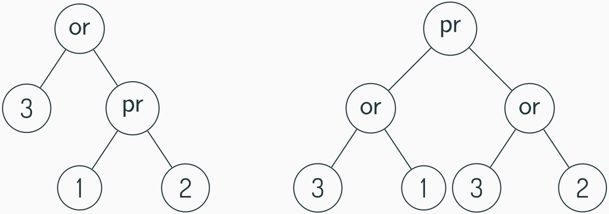
Idea

Denotational semantics provides admissibility and compatibility for free...

Probability and non-determinism

Bad operational preorder (not substitutive)

$$\forall H \subseteq \mathrm{Nat}, \sup_s \mathbb{P}(t/s \in H) \leq \sup_s \mathbb{P}(t'/s \in H)$$



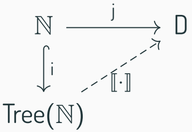
Hint: substitute using $1 \mapsto \frac{7}{8}\underline{0} + \frac{1}{8}\underline{1}$, $2 \mapsto 1$, $3 \mapsto \frac{3}{4}\underline{0} + \frac{1}{4}\underline{1}$ and compute for $H = \{1\}$ ...

The zen way of building contextual preorders

Nice, but what does this mean?

It really looks like denotational semantics...

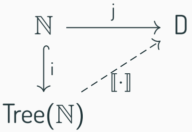
Given a continuous Σ-algebra D and a morphism $[\![\cdot]\!] : \mathbb{N}_\perp \to D$ one can build the preorder $\preccurlyeq_{\mathrm{den}}$.



$$t \preccurlyeq_{\mathrm{den}} t' \iff [\![t]\!] \leq_D [\![t']\!] \quad (1)$$

It really looks like denotational semantics...

Given a continuous Σ-algebra D and a morphism $[\![\cdot]\!] : \mathbb{N}_\perp \to D$ one can build the preorder $\preccurlyeq_{den}$.



$$t \preccurlyeq_{den} t' \iff [\![t]\!] \leq_D [\![t']\!] \qquad (1)$$

Properties of $\preccurlyeq_{den}$

1. Automatically admissible (continuity)
2. Automatically compatible (Σ-algebra)
3. Not always compositional! ($+, \times$ interpreted naturally)

Denotationally defined preorders

Factorisation

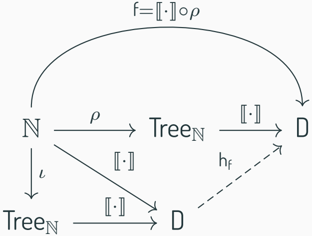
The map $[\![\cdot]\!] : \mathbb{N} \to D$ is said to have the factorisation property if, for every function $f : \mathbb{N} \to D$, there exists a continuous homomorphism $h_f : D \to D$ such that $f = h_f \circ [\![\cdot]\!]$.

$$\mathbb{N} \xrightarrow{[\![\cdot]\!]} D \overset{h_f}{\dashrightarrow} D, \qquad f : \mathbb{N} \to D$$

Consequence

We then have $[\![t\sigma]\!] = h_\sigma([\![t]\!])$ which is continuous in t with a fixed σ.

Proof



In practice...Monads

If (T, η, μ) is a monad over continuous Σ-algebras, the map $\eta \colon \mathsf{Nat} \to \mathsf{TNat}$ satisfies the factorisation property with $\mathsf{h_f} \triangleq \mathsf{f}^\dagger$.

Solution(s) for $\preccurlyeq^{\text{den}}$

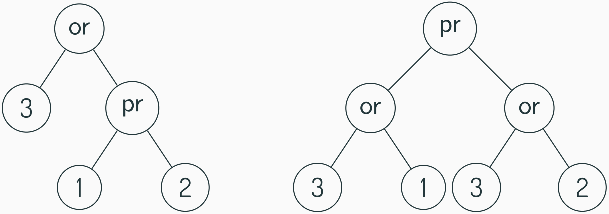
Kegelspitze Keimel and Plotkin (2017)

Scott closed convex subsets of valuations over X ordered by inclusion.

Previsions Goubault-Larrecq (2016)

$\mathcal{L}(X)$ is the set of lower semicontinuous maps from X to $\overline{\mathbb{R}}$. We use $\mathcal{L}(\mathcal{L}(X))$ to represent previsions.

Concretely



The zen way of building contextual preorders

Axiomatics

Let Vars be a set of countably many distinct variables

$$\left(\bigwedge_{i\in I} e_i \leq e_i'\right) \implies e \leq e' \quad ,$$

An effect theory T is a set of Horn clauses.

Order associated to a theory

There exists a smallest admissible and compositional preorder $\preccurlyeq_T$ satisfying T.

Concretely... $\preccurlyeq^{ax}$

Bot:	$\bot \leq x$
Prob:	$x \,\mathrm{pr}\, x = x,\; x \,\mathrm{pr}\, y = y \,\mathrm{pr}\, x,\; (x \,\mathrm{pr}\, y) \,\mathrm{pr}\, (z \,\mathrm{pr}\, w) = (x \,\mathrm{pr}\, z) \,\mathrm{pr}\, (y \,\mathrm{pr}\, w)$
Appr:	$x \,\mathrm{pr}\, y \leq y \implies x \leq y$
Nondet:	$x \,\mathrm{or}\, x = x,\; x \,\mathrm{or}\, y = y \,\mathrm{or}\, x,\; x \,\mathrm{or}\, (y \,\mathrm{or}\, z) = (x \,\mathrm{or}\, y) \,\mathrm{or}\, z$
Ang:	$x \,\mathrm{or}\, y \leq x$
Dem:	$x \,\mathrm{or}\, y \geq x$
Dist:	$x \,\mathrm{pr}\, (y \,\mathrm{or}\, z) = (x \,\mathrm{pr}\, y) \,\mathrm{or}\, (x \,\mathrm{pr}\, z)$

Figure 1: Horn theory for mixed probability and non determinism

Universal approximation scheme

Let $\frac{2^0-1}{2^0}t = \bot$ and $\frac{2^{(n+1)}-1}{2^{(n+1)}}t = t \text{ pr } \frac{2^n-1}{2^n}t$, extend with $\frac{2^\infty-1}{2^\infty}t = \bigsqcup_n \frac{2^n-1}{2^n}t$.

In a reasonable interpretation of trees $\frac{2^n-1}{2^n}t \ll t$ and $\frac{2^\infty-1}{2^\infty}t = t$.

Removing implications

For any effect theory containing the Bot and Prob axioms, an admissible model satisfies the Appr axiom if and only if it satisfies the equation $\frac{2^\infty-1}{2^\infty}x = x$.

The hard part $\preccurlyeq^{den} \subseteq \preccurlyeq^{ax}$

(i) Over trees without or: well-known since Heckmann (1994)

- Finite case: normal form
- Infinite case: approximation

(ii) Over trees with finitely many or

- Use distributivity to have a finite hat of or nodes
- $t' \equiv_{ax,den} t'$ or k, when $k = \lambda_1 t_1 + \cdots + \lambda_n t_n$
- $\forall i, \exists k_i := \lambda_1 t_1' + \cdots + \lambda_n t_n', t_i \preccurlyeq^{den} k_i$
- $t \preccurlyeq^{ax} t'$ or k_1 or $\cdots$ or $k_n \equiv_{ax,den} t'$

(iii) Over arbitrary trees using $\frac{2^n - 1}{2^n} t$, admissibility and the way-below relation

What admissibility? Why the way-below relation?

$$\left[\!\left[\frac{2^n-1}{2^n}t\right]\!\right] \ll [\![t]\!] \leq \bigsqcup_m \left[\!\left[\frac{2^m-1}{2^m}t'\right]\!\right]$$

Hence there exists an m such that

$$\left[\!\left[\frac{2^n-1}{2^n}t\right]\!\right] \leq \left[\!\left[\frac{2^m-1}{2^m}t'\right]\!\right]$$

And conclude using the approximation.

The zen way of building contextual preorders

Can we forget about domain theory?

Self-contained preorder definition

Given a strategy $s\colon \{l, r\}^* \to \{l, r\}$ and a tree t evaluate $t \restriction s$.

$$t \preccurlyeq^{op} t' \Leftrightarrow \forall h\colon \mathbb{N} \to [0, \infty] \ \sup_s E_{t \restriction s}(h) \leq \sup_s E_{t' \restriction s}(h)$$

1. Compatible: easy
2. Substitutive: easy
3. Admissible: the function $G_h\colon (s, t) \mapsto \mathbb{E}_{t \restriction s}(h)$ is continuous and the set of strategies is compact.

The correspondance between the operational preorder and the denotational one is observed through the isomorphism between $\mathcal{L}(\mathcal{L}(X))$ and $\mathcal{SV}_{\leq 1}X$ noticed by Keimel and Plotkin (2017) and Goubault-Larrecq (2016)

$$\Lambda\colon A \mapsto \left(f \mapsto \inf_{\mu \in A} \int_{n \in \mathbb{N}} f(n) d\mu \right)$$

Missing

What I did not tell

Probabilities and non-determinism are still a thing

- Call-by-push-value and full abstraction for PCF with probabilities and non-determinism Goubault-Larrecq (2019)
- Weak distributive laws allow to combine “naturally” probabilities and non-determinism Goy and Petrisan (2020)
- And many more!

Thank you ☺

Bauer, A. and Pretnar, M. (2012). Programming with algebraic effects and handlers. CoRR, abs/1203.1539.

Dal Lago, U., Gavazzo, F., and Blain Levy, P. (2017). Effectful Applicative Bisimilarity: Monads, Relators, and Howe's Method (Long Version). ArXiv e-prints.

Goubault-Larrecq, J. (2016). Isomorphism theorems between models of mixed choice. Mathematical Structures in Computer Science. To appear.

Goubault-Larrecq, J. (2019). A probabilistic and non-deterministic call-by-push-value language. In 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019, pages 1–13. IEEE.

Goy, A. and Petrisan, D. (2020). Combining probabilistic and non-deterministic choice via weak distributive laws. In Hermanns, H., Zhang, L., Kobayashi, N., and Miller, D., editors, LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020, pages 454–464. ACM.

Heckmann, R. (1994). Probabilistic domains. Trees in Algebra and Programming—CAAP'94, pages 142–156.

Johann, P., Simpson, A., and Voigtländer, J. (2010). A generic operational metatheory for algebraic effects. In 2010 25th Annual IEEE Symposium on Logic in Computer Science, pages 209–218.

Keimel, K. and Plotkin, G. D. (2017). Mixed powerdomains for probability and nondeterminism. Logical Methods in Computer Science, 13(1).

Pitts, A. M. (2000). Parametric polymorphism and operational equivalence. Mathematical. Structures in Comp. Sci., 10(3):321–359.

Plotkin, G. and Power, J. (2001). Adequacy for algebraic effects. In International Conference on Foundations of Software Science and Computation Structures, pages 1–24. Springer.

Plotkin, G. D. (1977). Lcf considered as a programming language. Theoretical computer science, 5(3):223–255.

Plotkin, G. D. and Pretnar, M. (2013). Handling Algebraic Effects. Logical Methods in Computer Science, Volume 9, Issue 4.

Wadler, P. (1989). Theorems for free! In Proceedings of the fourth international conference on Functional programming languages and computer architecture, pages 347–359. ACM.