Algebraic Effects

Gordon Plotkin

Laboratory for the Foundations of Computer Science, School of Informatics,
University of Edinburgh

CIRM, Marseille, February, 2012

Outline

Outline

1. Moggi's Monads As Notions of Computation
2. Algebraic Effects
   - Introduction
   - Equational theories
     - Finitary equational theories
     - Algebra with parameterised operations
     - Algebra with parameters and parametric arguments
     - Algebraic operations and generic effects
   - Continuous algebra
3. Discussion

The λ-calculus: syntax

Raw Syntax

Types $\quad \sigma ::= \sigma \to \tau$

Terms $\quad M ::= x \mid \lambda x : \sigma .\, M \mid MN$

Typing

Environments $\quad \Gamma ::= x_1 : \sigma_1, \ldots, x_n : \sigma_n$

Judgments $\quad \Gamma \vdash M : \sigma$

Rules $\quad \dfrac{\Gamma, x{:}\sigma \vdash M{:}\tau}{\Gamma \vdash \lambda x{:}\sigma .\, M{:}\sigma \to \tau}$

The λ-calculus: semantics in **Set**

Types $$[\![\sigma]\!] \in \mathbf{Set}$$

$$[\![\sigma \to \tau]\!] = [\![\sigma]\!] \Rightarrow [\![\tau]\!]$$

Environments $$[\![x_1 : \sigma_1, \ldots, x_n : \sigma_n]\!] = [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!]$$

Terms $$[\![\Gamma]\!] \xrightarrow{[\![M]\!]} [\![\tau]\!]$$

$$[\![\Gamma]\!] \xrightarrow{[\![\lambda x{:}\sigma.\, M]\!]} [\![\sigma \to \tau]\!] = \mathrm{Curry}([\![\Gamma, x : \sigma]\!] \xrightarrow{[\![M]\!]} [\![\tau]\!])$$

Adding recursion to the λ-calculus

Given

$$f : \sigma \to \tau, x : \sigma \vdash M : \tau$$

we would like to define f by:

$$f(x) = M$$

So we introduce a term for making such definitions:

$$\mathrm{rec}\ f : \sigma \to \tau, x : \sigma.\, M$$

$$\frac{\Gamma, f : \sigma \to \tau, x : \sigma. \vdash M : \tau}{\Gamma \vdash \mathrm{rec}\ f : \sigma \to \tau, x : \sigma.\, M : \sigma \to \tau}$$

For the semantics we use cpos.

Recap on **Cpo**

- A *cpo* P is a partial order with lubs $\bigvee_n x_n$ of increasing sequences $x_0 \leq x_1 \leq \ldots x_n \leq \ldots$
- A function $f : P \to Q$ between cpos is *continuous* if it is monotone and preserves lubs of increasing sequences, ie:

$$f(\bigvee_n x_n) = \bigvee_n f(x_n)$$

- $P \Rightarrow Q$ is the cpo of all such functions, ordered pointwise:

$$f \leq g \equiv \forall x \in P.\, f(x) \leq g(x)$$

- $P \times Q$ is also a cpo if ordered coordinatewise:

$$(x, y) \leq (x', y') \equiv x \leq x' \text{ and } y \leq y'$$

Pointed cpos

- A cpo P is *pointed* if it has a least element $\perp$ when it is a *cppo* (and $P \to Q$ is pointed if Q is).
- Every continuous endofunction f on a cppo D has a least (pre-)fixed point, given by:

$$\mathrm{fix}(f) = \bigvee_n f^n(\perp)$$

- Every $f : P \times D \to D$ has a parameterised fixed-point

$$f^\dagger : P \to D =_{\mathrm{def}} x \in P \mapsto \mathrm{fix}(f(x, \cdot))$$

- Every cpo P can be *lifted* to form a cppo $P_\perp = P \cup \{\perp\}$ with a new least element.

The λ-calculus: semantics in **Cpo**

Types $$[\![\sigma]\!] \in \mathbf{Cpo}$$

$$[\![\sigma \to \tau]\!] = [\![\sigma]\!] \Rightarrow [\![\tau]\!]_\perp$$

Environments $$[\![x_1 : \sigma_1, \ldots, x_n : \sigma_n]\!] = [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!]$$

Terms $$[\![\Gamma]\!] \xrightarrow{[\![M]\!]} [\![\tau]\!]_\perp$$

$$[\![\Gamma]\!] \xrightarrow{[\![\lambda x{:}\sigma.\ M]\!]} [\![\sigma \to \tau]\!] = \mathrm{Curry}([\![\Gamma, x : \sigma]\!] \xrightarrow{[\![M]\!]} [\![\tau]\!]_\perp)$$

The λ-calculus: semantics in **Cpo**: recursion

From

$$\Gamma, f : \sigma \to \tau, x : \sigma. \vdash M : \tau$$

we have, successively:

$$\begin{array}{c}
[\![\Gamma]\!] \times [\![\sigma \to \tau]\!] \times [\![\sigma]\!] \xrightarrow{[\![M]\!]} [\![\tau]\!] \\
[\![\Gamma]\!] \times [\![\sigma \to \tau]\!] \xrightarrow{\mathrm{Curry}([\![M]\!])} [\![\sigma \to \tau]\!] \\
[\![\Gamma]\!] \xrightarrow{\mathrm{Curry}([\![M]\!])^{\dagger}} [\![\sigma \to \tau]\!]
\end{array}$$

which is

$$[\![\Gamma]\!] \xrightarrow{\mathrm{rec}\ f:\sigma\to\tau, x:\sigma.\, M} [\![\sigma \to \tau]\!]$$

Moggi's insight

Going through the semantics in detail one needs some natural functions associated to lifting:

- Functorial action $\mathbf{Cpo}(P, Q) \xrightarrow{(\,\cdot\,)_\perp} \mathbf{Cpo}(P_\perp, Q_\perp)$

This makes lifting a functor

- Unit $P \xrightarrow{\eta} P_\perp$
- Multiplication $(P_\perp)_\perp \xrightarrow{\mu} P_\perp$

and then these make lifting a monad

- Strength $P \times Q_\perp \xrightarrow{\mathrm{st}} (P \times Q)_\perp$

and then this makes lifting a strong monad

Moggi's insight (cntnd.)

(Other) computational effects can also be modelled by monads T, e.g. in **Set**:

- Exceptions $T_{\mathrm{exc}}(X) = X + E$
- State $S \times X \longrightarrow S \times Y$ can be rewritten as $X \longrightarrow (S \times Y)^S$ and $T_{\mathrm{state}}(X) = (S \times X)^S$ is a strong monad.
- Finite Nondeterminism $T_{\mathrm{SL}}(X) = \mathcal{F}^+(X)$ the collection of non-empty finite subsets of X.
- Continuations $T_{\mathrm{cont}}(X) = R^{R^X}$

and there are many other examples, including combinations, such as this for state plus exceptions:

$$T(X) = (S \times (X + E))^S$$

In **Cpo** one has similar examples, generally including lifting to accommodate recursion, so that $T(P)$ is a cppo, e.g., for state plus nontermination: $T(P) = ((S \times P)_\perp)^S$

Moggi's computational λ-calculus: semantics in a ccc **C** with a strong monad T

Types

$$[\![\sigma]\!] \in \mathbf{C}$$

$$[\![\sigma \to \tau]\!] = [\![\sigma]\!] \Rightarrow T([\![\tau]\!])$$

Environments

$$[\![x_1 : \sigma_1, \ldots, x_n : \sigma_n]\!] = [\![\sigma_1]\!] \times \ldots \times [\![\sigma_n]\!]$$

Terms

$$[\![\Gamma]\!] \xrightarrow{[\![M]\!]} T([\![\tau]\!])$$

$$[\![\Gamma]\!] \xrightarrow{[\![\lambda x{:}\sigma.\, M]\!]} [\![\sigma \to \tau]\!] = \mathrm{Curry}([\![\Gamma, x : \sigma]\!] \xrightarrow{[\![M]\!]} T([\![\tau]\!]))$$

Outline

Outline

Two questions

- So we have a denotational semantics. How about an operational one?
- How do effects arise, i.e., how do we "construct" them in a programming language?

- Answering the first question immediately leads to the second.
- Answering that leads to understanding where Moggi's monads come from.

An example: finite nondeterminism

Working in **Set** we take $T_{\mathrm{SL}}(X) = \mathcal{F}^{+}(X)$ the collection of non-empty finite subsets of X.

To create the effects we add an *effect constructor*:

$$\frac{M : \sigma \quad N : \sigma}{M + N : \sigma}$$

Nondeterminism as an algebraic effect

There is a natural equational theory, with signature $+ : 2$, and set of axioms SL (for semilattices) given by:

$$\begin{array}{lccc} \textit{Associativity} & (x+y)+z & = & x+(y+z) \\ \textit{Commutativity} & x+y & = & y+x \\ \textit{Absorption} & x+x & = & x \end{array}$$

The evident algebra on $\mathcal{F}^{+}(X)$ satisfies these equations, interpreting $+$ as $\cup$.

Further:

$\mathcal{F}^{+}$ is the free algebra monad.

Is this the right set of axioms?

- An equational theory is *equationally inconsistent* if it proves $x = y$.
- An equational theory is *Hilbert-Post complete* if adding an unprovable equation makes it equationally inconsistent.

Theorem

ND is Hilbert-Post complete.

Proof.

Let $t = u$ be an unprovable equation, and assume it. Then there is a variable x in one of t or u, but not in the other. Equating all the other variables to y one obtains one of the following two equations: $x = y$ or $x + y = y$. One obtains $x = y$ from either of these. $\square$

Other effects

- Similar results hold in **Set** for, eg, exceptions, (global) state; I/O; (probabilistic) nondeterminism; and combinations thereof.
- May need infinitary algebra and parameterised operations.
- Works similarly for **Cpo** but also need inequations $t \leq u$.
- For other categories there is a general theory (of enriched Lawvere theories). Problem: categories of presheaves as applied to the treatment of new variables and fresh names.

Outline

Finitary equational theories: syntax

- Signature $\Sigma_e = (\mathrm{Op}, \mathrm{ar} : \mathrm{Op} \to \mathbb{N})$. We write $\mathrm{op} : n$ for arities.
- Terms $t ::= x \quad | \quad \mathrm{op}(t_1, \ldots, t_n)\ (\mathrm{op} : n)$. We leave open what the set Var of variables is.
- Equations $t = u$
- Axiomatisations Sets Ax of equations
- Deduction $\mathrm{Ax} \vdash t = u$
- Theories Sets of equations Th closed under deduction

Addition to λ-calculus syntax

$$\frac{\Gamma \vdash M_1 : \sigma, \ldots, \Gamma \vdash M_n : \sigma}{\Gamma \vdash \mathrm{op}(M_1, \ldots, M_n) : \sigma} \quad (\mathrm{op} : n)$$

Finitary equational theories: semantics

- Algebras $\mathcal{A} = (A, \mathrm{op}_{\mathcal{A}} : A^n \longrightarrow A \ \ (\mathrm{op} : n))$
- Homomorphisms $h : \mathcal{A} \to \mathcal{B}$ are functions $h : A \to B$ such that, for all $\mathrm{op} : n$, and $a_1, \ldots, a_n \in A$:

$$h(\mathrm{op}_{\mathcal{A}}(a_1, \ldots, a_n)) = \mathrm{op}_{\mathcal{B}}(h(a_1), \ldots, h(a_n))$$

- Denotation $\mathcal{A}[\![t]\!](\rho)$, where $\rho : \mathrm{Var} \to A$.
- Validity $\mathcal{A} \models t = u$
- Models $\mathcal{A}$ is a *model* of Ax if $\mathcal{A} \models t = u$, for all $t = u$ in Ax.

The free algebra monad T_{Ax} of an axiomatic theory Ax

The free model $F_{\mathrm{Ax}}(X)$ of Ax over a set X has carrier:

$$T_{\mathrm{Ax}}(X) =_{\mathrm{def}} \{[t]_{\mathrm{Ax}} \mid t \text{ is a term with variables in } X\}$$

where $[t]_{\mathrm{Ax}} =_{\mathrm{def}} \{u \mid \mathrm{Ax} \vdash u = t\}$.

Its operations are given by:

$$\mathrm{op}_{F_{\mathrm{Ax}}(X)}([t]_1, \ldots, [t]_n) = [\mathrm{op}(t_1, \ldots, t_n)] \quad (\mathrm{op} : n)$$

Freeness

For any model $\mathcal{A}$ of Ax, and any function $f : X \to A$ there is a unique homomorphism $f^\dagger : F_{\mathrm{Ax}}(X) \to \mathcal{A}$ such that the following diagram commutes:

$$\begin{tikzcd} X \arrow[d, "\eta"'] \arrow[dr, "f"] & \\ T_{\mathrm{Ax}}(X) \arrow[r, "f^\dagger"] & A \end{tikzcd}$$

where $\eta =_{\mathrm{def}} x \mapsto [x]_{\mathrm{Ax}}$

Remarks:

1. $f^\dagger([t]) = \mathcal{A}[\![t]\!](f)$
2. $T_{\mathrm{Ax}}(X)$ is a monad with unit η and multiplication $(\mathrm{id}_{T_{\mathrm{Ax}}(X)})^\dagger$.

Another example: exceptions

Given a (possibly infinite) set E of exceptions, the signature has nullary operation symbols:

$$\mathrm{raise}_e \quad (e \in E)$$

The set of axioms Exc is empty, and one obtains the usual exceptions monad

$$T_{\mathrm{Exc}}(X) = X + E$$

There is then a puzzle: how do exception handlers fit into the algebraic theory of effects - more later!

Yet another example: probabilistic computation

We have n-ary operations $\sum_{n,p_1,\ldots,p_n}$ for all $n \geq 0$ and n-tuples of non-negative reals $p_1, \ldots, p_n$ summing to 1. The set of axioms Con is

1. $\sum_{i=1,m} \delta^i_j x_i = x_j$
2. $\sum_{i=1,m} p_i \sum_{j=1,n} q_{ij} x_j = \sum_{j=1,n}(\sum_{i=1,m} p_i q_{ij}) x_j$

where $\delta^i_j = \begin{cases} 1 & (\text{if } i = j) \\ 0 & \text{otherwise} \end{cases}$

The monad is the set of finite probability distributions over X

$$T_{Con}(X) = \mathcal{D}_\omega(X) =_{\mathrm{def}} \{\sum_{i=1}^{n} \lambda_i x_i \mid n \geq 0, \lambda_i \geq 0, \sum_i \lambda_i = 1\}$$

and the unit is $\eta(x) = \delta_x$ the Dirac probability distribution on x.

Remarks on Con

- Con is not HP complete, but its only proper equational extension is SL, the theory of semilattices (this is a non-trivial result). These have an associative, commutative binary operator.
- Con can be equivalently axiomatised using probabilistic choice binary operators $+_p$, for $0 \leq p \leq 1$. In terms of Con these operators are defined by:

$$x +_p y = px + (1-p)y$$

Parametric finitary equational theories: syntax

- First-order multi-sorted signature

$$\Sigma_p = (S, \mathrm{Fun}, \mathrm{Pred}, \mathrm{ar}_{\mathrm{fun}} : \mathrm{Fun} \to S^* \times S, \mathrm{ar}_{\mathrm{pred}} : \mathrm{Pred} \to S^*)$$

- Parametric signature

$$\Sigma_e = (\mathrm{Op}, \mathrm{ar}_{\mathrm{op}} : \mathrm{Op} \to S^* \times \mathbb{N})$$

- Terms
  $t ::= x \quad | \quad \mathrm{op}_{u_1,\ldots,u_m}(t_1, \ldots, t_n)$ $(\mathrm{op} : s_1, \ldots s_m; n$ and $u_i : s_i)$.
- Equations $t = u \ \ (\varphi)$ where φ is a first-order formula over Σ_p.
- Axiomatisations Sets Ax of equations
- Deduction (an interesting question, not treated here)

Examples

- Exceptions Σ_p has a single sort exc, and constants $e : 0$ for each $e \in E$.
  Σ_e has a single operation symbol $\mathrm{raise} : \mathrm{exc}; 0$. There are no equations.
- Probability Σ_p has a single sort real, constants $0, 1$, binary function symbols $+, \times$, a unary function symbol $-$, and a relation symbol $\leq$.
  Σ_e has a single binary operation symbol $+ : \mathrm{real}; 2$. Here is an example equation:

$$+_p(x, y) = +_{1-p}(x, y) \ \ (0 \leq p \leq 1)$$

Addition to λ-calculus syntax

- Types

$$\sigma ::= s \ \ (s \in S) \mid \mathrm{bool}$$

- Terms

$$\begin{array}{lll} M & ::= & f(M_1, \ldots, M_n) \ \ (f \in \mathrm{Fun}) \mid P(M_1, \ldots, M_n) \ \ (P \in \mathrm{Pred}) \mid \\ & & \mathrm{tt} \mid \mathrm{ff} \mid \mathrm{if}\ L\ \mathrm{then\ M\ else}\ N \mid \\ & & \mathrm{op}_{M_1,\ldots,M_m}(N_1, \ldots, N_n) \end{array}$$

- Example type-checking rule

$$\frac{\Gamma \vdash M_1 : s_1, \ldots, \Gamma \vdash M_m : s_m, \ \ \Gamma \vdash N_1 : \sigma, \ldots, \Gamma \vdash N_n : \sigma}{\Gamma \vdash \mathrm{op}_{M_1,\ldots,M_m}(N_1, \ldots, N_n) : \sigma}$$

where $\mathrm{op} : s_1, \ldots, s_m; n$

Parametric finitary equational theories: semantics

- **Parameter interpretation** We fix an interpretation $\mathcal{M}$ of Σ_p.
- **Algebras** With that, a Σ_e-algebra is a structure

$$(A, \mathrm{op}_{\mathcal{A}} : \mathcal{M}[\![\mathbf{s}]\!] \times A^n \to A \quad (\mathrm{op} : \mathbf{s}; n))$$

  where $\mathcal{M}[\![s_1, \ldots, s_m]\!] =_{\mathrm{def}} \mathcal{M}[\![s_1]\!] \times \ldots \mathcal{M}[\![s_m]\!]$.
  Homomorphisms are then defined in the evident way.
- **Denotation** $\mathcal{A}[\![t]\!](\rho_p, \rho_e)$, where $\rho_e : \mathrm{Var} \to A$. For example

$$\mathcal{A}[\![\mathrm{op}_{u_1,\ldots,u_m}(t_1, \ldots, t_n)]\!](\rho_p, \rho_e) = \mathrm{op}_{\mathcal{A}}(\mathcal{M}[\![u_1, \ldots, u_m]\!](\rho_p), \mathcal{A}[\![t_1]\!](\rho_p, \rho_e), \ldots, \mathcal{A}[\![t_n]\!](\rho_p, \rho_e))$$

  Validity and **Models** are then defined in the evident way.

Free algebra theorem

Theorem

Let Ax *be a set of parametric* Σ_e*-axioms. Then there is a free model* $F_{\mathrm{Ax}}(X)$ *of* Ax *over any* X*. That is, there is a* $\eta : X \to T_{\mathrm{Ax}}(X)$*, where* $T_{\mathrm{Ax}}(X)$ *is the carrier of* $F_{\mathrm{Ax}}(X)$*, such that for any model* $\mathcal{A}$ *of* Ax*, and any function* $f : X \to A$ *there is a unique homomorphism* $f^\dagger : F_{\mathrm{Ax}}(X) \to \mathcal{A}$ *such that the following diagram commutes:*

$$\begin{tikzcd} X \arrow[d, "\eta"'] \arrow[dr, "f"] & \\ T_{\mathrm{Ax}}(X) \arrow[r, "f^\dagger"] & A \end{tikzcd}$$

Idea of proof

The idea is to reduce to ordinary equational theories.

- For every $\mathrm{op} : \mathbf{s}; n$ and $(a_1, \ldots, a_m) \in \mathcal{M}[\![\mathbf{s}]\!]$ we introduce an operation symbol $f_{a_1,\ldots,a_m} : n$.
- Then from any parametric term t and ρ_p we can obtain an ordinary term t^{ρ_p}. For example:

$$\mathrm{op}_{u_1,\ldots,u_m}(t_1, \ldots, t_n)^{\rho_p} = \mathrm{op}_{\mathcal{M}[\![u_1,\ldots,u_m]\!](\rho_p)}(t_1^{\rho_p}, \ldots, t_n^{\rho_p})$$

- Then one obtains a set of ordinary equations from any parametric equation in Ax, taking all ρ_p's.
- We know all these ordinary equations have a free model. That immediately gives a parametric model of Ax with the same carrier,"gluing" the interpretations of all the $f_{a_1,\ldots,a_m}$ together. Keeping the same unit, we immediately deduce parametric freeness from ordinary freeness.

State treated algebraically

Suppose we have locations which can store natural numbers. We have natural programming notation for reading and writing:

$$\frac{M : \mathrm{loc}}{!M : \mathrm{nat}} \qquad \frac{M : \mathrm{loc}, N : \mathrm{nat}}{M := N : \mathrm{unit}}$$

But

$$\mathrm{loc} \xrightarrow{\;!\;} \mathrm{nat} \qquad \mathrm{loc} \times \mathrm{nat} \xrightarrow{:=} \mathrm{unit}$$

do not seem to have much to do with algebra.

Hint: Read "$M + N$" as "choose 0 or 1 and then do whichever continuation M or N is appropriate."

One can read $M +_p N$ similarly, but in terms of tossing a biased coin with head having probability p.

State treated algebraically (cntnd.)

So for writing we would have an operation, update say, which writes and then carries on (i.e. has a single continuation). This suggests:

$$\mathrm{update} : \mathrm{loc}, \mathrm{nat}; 1$$

which fits within parametric algebra.
For reading we would have an operation, lookup say, which reads a location and then carries on with a continuation depending on the value read. This suggests:

$$\mathrm{lookup} : \mathrm{loc}; \mathrm{nat}$$

a parameterised infinitary operation!
So we now look at infinitary algebra and a finitary notation for it.
We will return later to the status of things like ! and $:=$ and see that they form part of a general pattern of generic effects.

Infinitary equational logic: syntax

- Signature $\Sigma_e = (\mathrm{Op}, \mathrm{ar} : \mathrm{Op} \to \omega + 1)$. We write $\mathrm{op} : n$ for arities, including ω.
- Terms as in finitary case plus: $\mathrm{op}(t_1, t_2, \ldots, t_n, \ldots)$ $(\mathrm{op} : \omega)$. We leave open what the set Var of variables is.
- Equations $t = u$ as before
- Axiomatisations Sets Ax of equations
- Deduction $\mathrm{Ax} \vdash t = u$ an easy variant of the finitary case
- Theories Sets of equations Th closed under deduction

Infinitary equational theories: semantics

Algebras are structures $\mathcal{A} = (A, \mathrm{op}_{\mathcal{A}} : A^n \longrightarrow A \ \ (\mathrm{op} : n))$, and recall that here n can be ω.

Homomorphisms $h : \mathcal{A} \to \mathcal{B}$ are, much as before, functions $h : A \to B$ such that, for all $\mathrm{op} : n$, and $\mathbf{a} \in A^n$:

$$h(\mathrm{op}_{\mathcal{A}}(\mathbf{a})) = \mathrm{op}_{\mathcal{B}}(h(\mathbf{a}))$$

Denotation $\mathcal{A}[\![t]\!](\rho)$ is also defined much as before.
Validity $\mathcal{A} \models t = u$ is defined as before.
Models $\mathcal{A}$ is also defined as before.

The free algebra monad T_{Ax} of an infinitary axiomatic theory Ax

All is as before. The free model $F_{\mathrm{Ax}}(X)$ over a set X has carrier:

$$T_{\mathrm{Ax}}(X) =_{\mathrm{def}} \{[t]_{\mathrm{Ax}} \mid t \text{ is a term with variables in } X\}$$

where $[t]_{\mathrm{Ax}} =_{\mathrm{def}} \{u \mid \mathrm{Ax} \vdash u = t\}$; its operations are given by:

$$\mathrm{op}_{F_{\mathrm{Ax}}(X)}([\mathbf{t}]) = [\mathrm{op}(\mathbf{t})] \quad (\mathrm{op} : n)$$

the unit $\eta : X \to T_{\mathrm{Ax}}(X)$ is again $x \mapsto [x]$; for any model $\mathcal{A}$ of Ax, and any function $f : X \to A$ the unique mediating homomorphism $f^{\dagger} : F_{\mathrm{Ax}}(X) \to \mathcal{A}$ is given by:

$$f^{\dagger}([t]) = \mathcal{A}[\![t]\!](f)$$

and the multiplication is $(\mathrm{id}_{T_{\mathrm{Ax}}(X)})^{\dagger}$.

Notation and equations for state

$$t ::= \mathrm{update}_{u_1,u_2}(t) \mid \mathrm{lookup}_u(n : \mathrm{nat}.\, t)$$

Equations for writing and reading a single location:

$$\begin{array}{lcll}
\mathrm{update}_{l,m}(\mathrm{update}_{l,n}(x)) & = & \mathrm{update}_{l,n}(x) & (1) \\
\mathrm{lookup}_l(m : \mathrm{nat}.\, \mathrm{lookup}_l(n : \mathrm{nat}.\, x(m,n))) & = & & \\
 & & \mathrm{lookup}_l(m : \mathrm{nat}.\, x(m,m)) & (2) \\
\mathrm{lookup}_l(n : \mathrm{nat}.\, x) & = & x & (3) \\
\mathrm{update}_{l,m}(\mathrm{lookup}_l(n : \mathrm{nat}.\, x(n))) & = & \mathrm{update}_{l,m}(x(m)) & (4) \\
\mathrm{lookup}_l(n : \mathrm{nat}.\, \mathrm{update}_{l,n}(x)) & = & x & (5)
\end{array}$$

Notation and equations for state (cntnd)

Commutation Equations for different locations

$$\mathrm{update}_{l,m}(\mathrm{update}_{l',n}(x)) = \mathrm{update}_{l',n}(\mathrm{update}_{l,m}(x)) \quad (l \neq l') \tag{7}$$

$$\begin{aligned}\mathrm{lookup}_l(m : \mathrm{nat}.\, \mathrm{lookup}'_l(n : \mathrm{nat}.\, x(m,n))) &= \\ &\mathrm{lookup}'_l(n : \mathrm{nat}.\, \mathrm{lookup}_l(m : \mathrm{nat}.\, x(m,n))) \quad (l \neq l')\end{aligned} \tag{8}$$

$$\begin{aligned}\mathrm{update}_{l,m}(\mathrm{lookup}'_l(n : \mathrm{nat}.\, x(n))) &= \\ &\mathrm{lookup}'_l(n : \mathrm{nat}.\, \mathrm{update}_{l,m}(x(n)))\end{aligned} \tag{9}$$

Redundancies

Equations (3), and (2) and (8) (Mellies) are redundant.
For example, for (3) we have:

$$\begin{array}{rcll} \mathrm{lookup}_l(n : \mathrm{nat}.\, x) & = & \mathrm{lookup}_l(n : \mathrm{nat}.\, \mathrm{update}_{l,n}(\mathrm{lookup}_l(n : \mathrm{nat}.\, x))) & \text{(by (5))} \\ & = & \mathrm{lookup}_l(n : \mathrm{nat}.\, \mathrm{update}_{l,n}(x))) & \text{(by (4))} \\ & = & x & \text{(by (5))} \end{array}$$

Parametric axiom. ths. with abstraction: syntax

- First-order multi-sorted signature

$$\Sigma_p = (S, \mathrm{Fun}, \mathrm{Pred}, \mathrm{ar}_{\mathrm{fun}} : \mathrm{Fun} \to S^* \times S, \mathrm{ar}_{\mathrm{pred}} : \mathrm{Pred} \to S^*, S_a)$$

  with a subcollection $S_a \subseteq S$ of *arity* sorts
- Parametric signature

$$\Sigma_e = (\mathrm{Op}, \mathrm{ar}_{\mathrm{op}} : \mathrm{Op} \to S^* \times A^{**})$$

- Terms

$$\frac{\Gamma, \mathbf{u} : \mathbf{s},\ \Gamma, \mathbf{x}_1 : \mathbf{s}_1 \vdash t_1, \ldots, \Gamma, \mathbf{x}_n : \mathbf{s}_n \vdash t_n}{\Gamma \vdash \mathrm{op}_{\mathbf{u}}(\mathbf{x}_1 : \mathbf{s}_1 . t_1, \ldots, \mathbf{x}_n : \mathbf{s}_n . t_n)} \qquad (\mathrm{op} : \mathbf{s}; \mathbf{s}_1, \ldots, \mathbf{s}_n)$$

- Equations $t = u \ \ (\varphi)$ and axiomatisations Ax are as before, and deduction remains an interesting question.

Addition to λ-calculus syntax

- Types

$$\sigma ::= s \;\; (s \in S) \mid \mathrm{bool}$$

- Terms

$$M \quad ::= \quad \mathrm{op}_{\mathbf{M}}(\mathbf{x}_1 : \mathbf{s}_1 . N_1, \ldots, \mathbf{x}_n : \mathbf{s}_n . N_n)$$

- Example type-checking rule

$$\frac{\Gamma \vdash \mathbf{M} : \mathbf{s}, \quad \Gamma, \mathbf{x}_1 : \mathbf{s}_1 \vdash N_1 : \sigma, \ldots, \Gamma, \mathbf{x}_n : \mathbf{s}_n \vdash N_n : \sigma}{\Gamma \vdash \mathrm{op}_{\mathbf{M}}(\mathbf{x}_1 : \mathbf{s}_1 . N_1, \ldots, \mathbf{x}_n : \mathbf{s}_n . N_n) : \sigma}$$

where $\mathrm{op} : \mathbf{s}; \mathbf{s}_1, \ldots, \mathbf{s}_m$

Parametric axiom. ths. with abstraction: semantics

- Parameter interpretation We fix an interpretation $\mathcal{M}$ of Σ_p, such that $\mathcal{M}[\![s]\!]$ is countable for all $s \in S_a$.
- Algebras With that, a Σ_e-algebra is a structure

$$(A, \mathrm{op}_{\mathcal{A}} : \mathcal{M}[\![\mathbf{s}]\!] \times A^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times A^{\mathcal{M}[\![\mathbf{s}_n]\!]} \to A \quad (\mathrm{op} : \mathbf{s}; \mathbf{s}_1, \ldots, \mathbf{s}_n))$$

- Denotation $\mathcal{A}[\![t]\!](\rho_p, \rho_e)$, where $\rho_e : \mathrm{Var} \to A$. For example

$$\mathcal{A}[\![\mathrm{op}_{\mathbf{u}}(\mathbf{x}_1 : \mathbf{s}_1.\, t_1, \ldots, \mathbf{x}_n : \mathbf{s}_n.\, t_n)]\!](\rho_p, \rho_e) = \mathrm{op}_{\mathcal{A}}(\mathcal{M}[\![\mathbf{u}]\!](\rho_p), \varphi_1, \ldots, \varphi_n)$$

where:

$$\varphi_i(\mathbf{a}_i) =_{\mathrm{def}} \mathcal{A}[\![t_i]\!](\rho_p[\mathbf{a}/\mathbf{x}_i], \rho_e) \qquad (i = 1, n,\ \mathbf{a}_i \in \mathcal{M}[\![s_i]\!])$$

Homomorphisms, Validity and Models are defined in the evident way.

Free algebras, etc.

- As usual, there is a free algebra $F_{\mathrm{Ax}}(X)$ over any set X, which induces the corresponding monad $T_{\mathrm{Ax}}(X)$.
- The proof is by a (now) evident reduction to (countably) infinitary equational logic.
- Restricting the denotations of arity types to be finite still covers many situations, e.g., locations storing bits or words. Thus abstraction can be useful even in the finitary case.

Side effects

- First order part The sorts are loc,nat, and there is a predicate symbol $=: \mathrm{loc}, \mathrm{loc}$. We assume $\mathcal{M}[\![=]\!]$ is equality, $\mathcal{M}[\![\mathrm{loc}]\!]$ is finite, and $\mathcal{M}[\![\mathrm{nat}]\!] = \mathbb{N}$. Set $\mathrm{Loc} =_{\mathrm{def}} \mathcal{M}[\![\mathrm{loc}]\!]$.
- Axioms Ax_S is as above.
- Monad $T_S(X) = (S \times X)^S$, where $S =_{\mathrm{def}} \mathbb{N}^{\mathrm{Loc}}$
- Operations

Lookup $\mathrm{Loc} \times T_S(X)^{\mathbb{N}} \xrightarrow{\mathrm{lookup}_{F_S(X)}} T_S(X)$ is defined by:

$$\mathrm{lookup}_{F_S(X)}(l, \varphi) = \sigma \mapsto \varphi(\sigma(l))$$

Update $\mathrm{Loc} \times \mathbb{N} \times T_S(X) \xrightarrow{\mathrm{update}_{F_S(X)}} T_S(X)$ is defined by:

$$\mathrm{update}_{F_S(X)}(l, n, \gamma) = \sigma \mapsto \gamma(\sigma[n/l])$$

Another example: interactive I/O

- First-order part The sorts are in, out; rest, including $\mathcal{M}$, as suits the purpose at hand.
- Operation symbols $\mathrm{input} : \varepsilon; \mathrm{in}$ and $\mathrm{output} : \mathrm{out}; 1$
- Axioms None!
- Monad $T_{I/O}(X)$ is the least set Y such that:

$$Y = Y^{\mathcal{M}[\![\mathrm{in}]\!]} + (\mathcal{M}[\![\mathrm{out}]\!] \times Y) + X$$

  and we just write:

$$T_{I/O}(X) = \mu Y.\ Y^{\mathcal{M}[\![\mathrm{in}]\!]} + (\mathcal{M}[\![\mathrm{out}]\!] \times Y) + X$$

- $T_{I/O}(X)$ is a collection of trees. Its internal nodes are either input ones, when they have an $\mathcal{M}[\![\mathrm{in}]\!]$-indexed collection of children, or output nodes, when they have an $\mathcal{M}[\![\mathrm{out}]\!]$ label and one child. Its leaves have an X label.

I/O cntnd.

- Operations

  Input $T_{I/O}(X)^{\mathcal{M}[\![\mathrm{in}]\!]} \xrightarrow{\mathrm{input}_{F_{I/O}(X)}} T_{I/O}(X)$ is defined by:

$$\mathrm{input}_{F_{I/O}(X)}(\varphi) = \mathrm{in}_1(\varphi)$$

  Output $\mathcal{M}[\![\mathrm{out}]\!] \times T_{I/O}(X) \xrightarrow{\mathrm{output}_{F_{I/O}(X)}} T_{I/O}(X)$ is defined by:

$$\mathrm{output}_{F_{I/O}(X)}(d, \gamma) = \mathrm{in}_2(d, \gamma)$$

The general case, when there are no axioms

We have:

$$T_{I/O}(X) = \mu Y. \sum_{\mathrm{op}:\mathbf{s};\mathbf{s}_1,\ldots,\mathbf{s}_n} (\mathcal{M}[\![\mathbf{s}]\!] \times Y^{\mathcal{M}[\![\mathbf{s}_1]\!] \times \ldots \times \mathcal{M}[\![\mathbf{s}_n]\!]}) + X$$

We again have a collection of trees. The internal nodes are $\mathcal{M}[\![\mathbf{s}]\!]$-labelled and have an $\mathcal{M}[\![\mathbf{s}_1]\!] \times \ldots \times \mathcal{M}[\![\mathbf{s}_n]\!]$-indexed collection of children. As before, the terminal nodes are X-labelled.

Algebraic operations

Fix a finitary equational axiomatic theory Ax. Then for any set X and operation symbol $\mathrm{op} : n$ we have the function:

$$T_{\mathrm{Ax}}(X)^n \xrightarrow{\mathrm{op}_{F_{\mathrm{Ax}}(X)}} T_{\mathrm{Ax}}(X)$$

Further for any function $f : X \to T_{\mathrm{Ax}}(Y)$, $f^\dagger$ is a homomorphism:

$$\begin{tikzcd}
T_{\mathrm{Ax}}(X)^n \arrow[r, "\mathrm{op}_{F_{\mathrm{Ax}}(X)}"] \arrow[d, "(f^\dagger)^n"'] \arrow[dr, phantom, "="] & T_{\mathrm{Ax}}(X) \arrow[d, "f^\dagger"] \\
T_{\mathrm{Ax}}(Y)^n \arrow[r, "\mathrm{op}_{F_{\mathrm{Ax}}(Y)}"'] & T_{\mathrm{Ax}}(Y)
\end{tikzcd}$$

We call such a famiy of functions $T_{\mathrm{Ax}}(X)^n \xrightarrow{\varphi_X} T_{\mathrm{Ax}}(X)$ algebraic.

Generic effects

- Given an algebraic family $T_{\mathrm{Ax}}(X)^n \xrightarrow{\varphi_X} T_{\mathrm{Ax}}(X)$, regarding n as $\{0, \ldots, n-1\}$, we obtain the generic effect:

$$e \in T_{\mathrm{Ax}}(n) = \varphi_n(\eta_n)$$

- Given $e \in T_{\mathrm{Ax}}(n)$ we obtain such an algebraic family by setting:

$$\varphi_X =_{\mathrm{def}} T_{\mathrm{Ax}}(X)^n \xrightarrow{(\,\cdot\,)^{\dagger}} T_{\mathrm{Ax}}(X)^{T_{\mathrm{Ax}}(n)} \xrightarrow{\;\cdot\,(e)\;} T_{\mathrm{Ax}}(X)$$

- This correspondence is a bijection between algebraic families and generic effects.
- Noting that $T_{\mathrm{Ax}}(X)^n$ is the collection of (equivalence classes) of terms with n free variables, we see (following the above definition) that the algebraic families are exactly the definable ones.

Examples

- Nondeterminism Corresponding to $+$ we have:

$$\mathrm{arb} \in T_{\mathrm{SL}}(\{0,1\}) = \{0,1\}$$

  which can be thought of as the (equivalence class of) the term $x + y$.

- Probabilistic nondeterminism Corresponding to $+$ we have:

$$\mathrm{coin}_p \in T_{\mathrm{SL}}(\{0,1\}) = p\delta_0 + (1-p)\delta_1$$

  which can be thought of as the (equivalence class of) the term $x +_p y$.

- Exceptions Roughly $\mathrm{raise}_e : 0$ is its own generic effect. Precisely, to the family $(\mathrm{raise}_e)_{\cdot+E} : \mathbb{1} = (X+E)^0 \to X + E$ corresponds $\mathrm{inr}(e) \in \emptyset + X$, which we can identify as e.

Algebraic operations, more generally

Fix a parametric equational axiomatic theory with abstraction Ax, and model $\mathcal{M}$. Then for any set X and operation symbol $\mathrm{op} : \mathbf{s}; \mathbf{s}_1, \ldots, \mathbf{s}_m$ we have the function:

$$\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_n]\!]} \xrightarrow{\mathrm{op}_{F_{\mathrm{Ax}}(X)}} T_{\mathrm{Ax}}(X)$$

Further for any function $f : X \to T_{\mathrm{Ax}}(Y)$, $f^{\dagger}$ is a homomorphism:

$$\begin{tikzcd}
\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_n]\!]} \arrow[r, "\mathrm{op}_{F_{\mathrm{Ax}}(X)}"] \arrow[d, swap, "\mathrm{id}_{\mathcal{M}[\![\mathbf{s}]\!]} \times (f^{\dagger})^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times (f^{\dagger})^{\mathcal{M}[\![\mathbf{s}_n]\!]}"] \arrow[dr, phantom, "="] & T_{\mathrm{Ax}}(X) \arrow[d, "f^{\dagger}"] \\
\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(Y)^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times T_{\mathrm{Ax}}(Y)^{\mathcal{M}[\![\mathbf{s}_n]\!]} \arrow[r, swap, "\mathrm{op}_{F_{\mathrm{Ax}}(Y)}"] & T_{\mathrm{Ax}}(Y)
\end{tikzcd}$$

We again call such a family of functions φ_X algebraic.

Generic effects, more generally

- Given an algebraic family

$$\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]} \times \ldots \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_n]\!]} \xrightarrow{\varphi_X} T_{\mathrm{Ax}}(X)$$

  equivalently: $\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]+\ldots+\mathcal{M}[\![\mathbf{s}_n]\!]} \xrightarrow{\varphi_X} T_{\mathrm{Ax}}(X)$
  we obtain the generic effect:

$$\mathcal{M}[\![\mathbf{s}]\!] \xrightarrow{e} T_{\mathrm{Ax}}(\mathcal{M}[\![\mathbf{s}_1]\!] + \ldots + \mathcal{M}[\![\mathbf{s}_n]\!]) = \varphi_{\sum_i \mathcal{M}[\![\mathbf{s}_i]\!]}(\,\cdot\,, \eta_{\sum_i \mathcal{M}[\![\mathbf{s}_i]\!]})$$

- Given such an e we obtain such an algebraic family:

$$\begin{array}{rcl} \mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\sum_i \mathcal{M}[\![\mathbf{s}_i]\!]} & \xrightarrow{\mathrm{id}_{\mathcal{M}[\![\mathbf{s}]\!]} \times (\,\cdot\,)^{\dagger}} & \mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{T_{\mathrm{Ax}}(\sum_i \mathcal{M}[\![\mathbf{s}_i]\!])} \\ & \xrightarrow{e \times \mathrm{id}} & T_{\mathrm{Ax}}(\sum_i \mathcal{M}[\![\mathbf{s}_i]\!]) \times T_{\mathrm{Ax}}(X)^{T_{\mathrm{Ax}}(\sum_i \mathcal{M}[\![\mathbf{s}_i]\!])} \\ & \xrightarrow{\mathrm{ev}} & T_{\mathrm{Ax}}(X) \end{array}$$

- This correspondence is a bijection between algebraic families and generic effects.

An example: side-effects

- Lookup The generic effect corresponding to

$$\mathrm{Loc} \times T_S(X)^{\mathbb{N}} \xrightarrow{\mathrm{lookup}_{F_S(X)}} T_S(X)$$

is

$$\mathrm{Loc} \xrightarrow{!} T_S(\mathbb{N}) = (S \times \mathbb{N})^S$$

where

$$!(l) = \sigma \mapsto (\sigma, \sigma(l))$$

- Update The generic effect corresponding to

$$\mathrm{Loc} \times \mathbb{N} \times T_S(X) \xrightarrow{\mathrm{update}_{F_S(X)}} T_S(X)$$

is

$$Loc \times \mathbb{N} \xrightarrow{:=} T_S(\mathbb{1})$$

where

$$:= (l, v) = \sigma \mapsto (\sigma[l/n], *)$$

Another example: interactive I/O

- Input The generic effect corresponding to

$$T_{I/O}(X)^{\mathcal{M}[\![\mathrm{in}]\!]} \xrightarrow{\mathrm{input}_{F_{I/O}(X)}} T_{I/O}(X)$$

  is

$$\mathrm{myread} \in T_{I/O}(\mathcal{M}[\![\mathrm{in}]\!])$$

  where

$$\mathrm{myread} = \mathrm{in}_1(d \in \mathcal{M}[\![\mathrm{in}]\!] \mapsto \mathrm{in}_3(d))$$

- Output The generic effect corresponding to

$$\mathcal{M}[\![\mathrm{out}]\!] \times T_{I/O}(X) \xrightarrow{\mathrm{output}_{F_{I/O}(X)}} T_{I/O}(X)$$

  is

$$\mathcal{M}[\![\mathrm{out}]\!] \xrightarrow{\mathrm{write}} T_{I/O}((1)$$

  where

$$\mathrm{write}(d) = \mathrm{in}_3(d, *)$$

Programming counterpart of being algebraic

- Evaluation contexts are given by:

$$\mathcal{E} ::= [\cdot] \mid \mathcal{E}N \mid (\lambda x : \sigma.\, M)\mathcal{E}$$

- For any operation symbol $\mathrm{op} : n$ we have:

$$\models \mathcal{E}[\mathrm{op}(M_1, \ldots, M_n)] = \mathrm{op}(\mathcal{E}[M_1], \ldots, \mathcal{E}[M_n])$$

  For example,

$$\models (M +_p M')N = (MN) +_p (M'N)$$

- More generally, for any operation symbol $\mathrm{op} : \mathbf{s}; \mathbf{s}_1, \ldots, \mathbf{s}_m$ we have:

$$\models \mathcal{E}[op_{\mathbf{M}}(\mathbf{x}_1 : \mathbf{s}_1.\, N_1, \ldots, \mathbf{x}_n : \mathbf{s}_n.\, N_n)] = op_{\mathbf{M}}(\mathbf{x}_1 : \mathbf{s}_1.\, \mathcal{E}[N_1], \ldots, \mathbf{x}_n : \mathbf{s}_n.\, \mathcal{E}[N_n])$$

  assuming variable clashes are avoided.

Outline

Example (in)equational theory: nontermination

The axiomatic theory Ax_Ω has a constant Ω and one axiom for it:

$$\Omega \leq x$$

The free-algebra monad is just lifting: $T(P) = P_\perp$

Another example: nondeterminism

As before we have a binary operation symbol op and the axioms SL of a semilattice. This is not order-HP complete but has two order-consistent extensions, given by these two axioms, respectively:

$$x \leq x + y$$

$$x \geq x + y$$

The two axiomatic theories are called SL_l and SL_u. The free continuous algebra for SL is the convex (aka Plotkin) powerdomain; that for SL_l is the lower (aka Hoare) powerdomain and that for SL_u is the upper (aka Smyth) powerdomain.

Finitary inequational theories: syntax

- Signature and terms are as before.
- Inequations $t \leq u$
- Axiomatisations Sets Ax of equations, as before.

Finitary inequational theories: semantics

- Algebras $\mathcal{A} = (A, \mathrm{op}_{\mathcal{A}} : A^n \longrightarrow A \ \ (\mathrm{op} : n))$ as before, except that A is a cpo and the $\mathrm{op}_{\mathcal{A}}$ are continuous.
- Homomorphisms as before, but assumed to be continuous.
- Denotation, validity, as before.
- Models $\mathcal{A}$ is a *model* of Ax if $\mathcal{A} \models t \leq u$, for all $t = u$ in Ax.
- Free models The free model $F_{\mathrm{Ax}}(P)$ of Ax over a cpo P always exists. T_{Ax} is strong (and has just one strength). In case Ax includes Ax_{Ω} the free model has a least element.

Parametric finitary inequational theories: syntax

- First-order multi-sorted signature and parametric signature are as before.
- Inequations $t \leq u \ \ (\varphi)$ where φ is a first-order formula over Σ_p.
- Axiomatisations as before.

Parametric finitary inequational theories: semantics

- Parameter interpretation We fix an interpretation $\mathcal{M}$ of Σ_p, allowing the denotations of sorts to be cpos, the denotations of function symbols are required to be continuous, and that of predicate symbols to be mediated by continuous functions.
- Algebras Σ_e-algebras and homomorphisms are as before, but again imposing continuity.
- Denotation, validity, and models are then defined in the evident way.
- Free models again exist (not by a reduction to the previous case) and the monad is strong and includes bottom if Ax_Ω is included in Ax.

Parametric axiom. ineq. ths. with abstraction

- First-order multi-sorted signature with a subcollection $S_a \subseteq S$ of arity sorts, as before. Parametric signature and terms as before.
- Equations $t \leq u \ \ (\varphi)$; then axiomatisations Ax as before.
- Parameter Interpretation As before, allowing cpos, imposing continuity, but now asking that arity sorts denote countable discrete cpos.
- Algebras, denotation, homomorphisms, validity, and models are defined in the evident way, imposing continuity.
- Free models again exist (e.g., by a reduction to countably infinitary continuous algebras with continuous parameters) and the monad is strong and includes bottom if Ax_Ω is included in Ax.

Some examples

- Exceptions + Ω $T(P) = (P + E)_\perp$
- State + Ω $T(P) = (S \times P)_\perp^S$
- I/O + Ω $T(P) =$ the initial cpo Q such that:

  $$Q \cong (Q^{\mathcal{M}[\![\mathrm{in}]\!]} + (\mathcal{M}[\![\mathrm{out}]\!] \times Q) + P)_\perp$$

  and we just write:

  $$T_{I/O}(P) = \mu Q.\, Q^{\mathcal{M}[\![\mathrm{in}]\!]} + (\mathcal{M}[\![\mathrm{out}]\!] \times Q) + P$$

- As a set, $T_{I/O}(P)$ is a collection of trees. Its internal nodes are either input ones, when they have an $\mathcal{M}[\![\mathrm{in}]\!]$-indexed collection of children, or output nodes, when they have an $\mathcal{M}[\![\mathrm{out}]\!]$ label and one child. Its leaves are labelled by either a P element or else by Ω.

Algebraic families and generic effects

The theory of these proceeds exactly analogously to before, now imposing the expected continuity conditions. No general assertions about definability are made as we have given no general pictures of the monads that arise, even for finitary equational theories, without parameters.

Outline

Remarks on generality

- Moving from **Set** to **Cpo** was tedious, and one fears the next case (presheaves, or presheaves over **Cpo**).
- Much of the theory can be developed generally for a category **V** (such as **Set** or **Cpo**) locally countably presentable as a cartesian closed category. One replaces equational theories with **V**-enriched Lawvere theories.These latter correspond exactly to strong monads on **V** with countable rank.
- However, for programming and logic, one still needs syntax so one in any case needs eventually to get concrete.
- Of course, to do all this one has to get to grips with enrichment and Lawvere theories.....

Is there a "logical" treatment of computational effects?

That is, as it seems, is there a treatment of computational effects following the Curry-Howard propositions-as-types view?

- Cartesian closed categories with a monad correspond to intuitionistic $\wedge$ and $\supset$ and a modality O (Fairtlough & Mendlers' lax logic), and, as a type theory, Moggi's computational metalanguage.
- But this is too external. Just taking Moggi's computational λ-calculus, one would get a strange (categorical) logic.
- Looking at Levy's CBPV seems like an interesting possibility. There are then two kinds of propositions, one corresponding to values, and one to computations. (This may remind one of polarised linear logic.)

Is there a "logical" treatment of computational effects (cntnd.)?

- But how would effect constructors (see below) fit within such a picture?
  In the lax setting, to each generic

  $$e : P \rightarrow T(I_1 + \ldots + I_n)$$

  one could associate an axiom

  $$P \Rightarrow \mathrm{O}(I_1 \vee \ldots \vee I_n)$$

  But for the proof-theoretic interpretation one would need to inject the relevant equivalences between proofs which would come from the equational axioms. So one feels little would be gained.
- Perhaps there is just some other Curry-Howard way altogether of thinking about effects – or at least some particular effects (other than continuations).

Some things that have been done so far

- Calculi with effects, such as λ_c and CBPV. (Moggi; Levy; Egger, Mogelberg & Simpson)
- ✓ (Moderately) general operational semantics. May not always get expected op. sems., eg, state. (Plotkin & Power; Power & Shkaravska)
- Work on general notions of observation and full abstraction. (Johann, Simpson & Voigtländer)
- ✓ Theory, and application, of effect deconstructors, such as exception handlers via not necessarily free algebras. (Plotkin & Pretnar; Plotkin & van Glabbeek)
- ✓ Combining monads in terms of combining theories, primarily sum and tensor. (Hyland,Plotkin & Power)
- Work on combining algebraic effects with continuations which are not algebraic and require special treatment. (Hyland, Levy, Power & Plotkin)
- First thoughts on a general logic of effects; connects with modal logic. Does not give Hoare logic. (Plotkin & Pretnar)
- ✓ Type and effect systems. (Kammar & Plotkin,Katsumata)
- Work on locality and effects (Melliès; Plotkin & Power; Power; Staton).