Lecture 1: Algebraic Effects I

Gordon Plotkin

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

NII Shonan Meeting No. 146
Programming and Reasoning with Algebraic Effects and Effect Handlers

Course

- Lecture 1: Algebraic effects I
- Lecture 2: Type and effect systems
- Lecture 3: Algebraic effects II
- Lecture 4: Effect handlers

Outline

Outline

The typed λ-calculus: syntax

Raw Syntax

Types $\quad \sigma ::= \mathrm{b} \mid \sigma \to \tau$

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

Typing

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

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

The typed λ-calculus: typing rules

Variables

$$x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash x_i : \sigma_i \quad (1 \leq i \leq n)$$

Constants

$$\Gamma \vdash c : \sigma \quad \text{(as given)}$$

Abstractions

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \to \tau}$$

Applications

$$\frac{\Gamma \vdash M : \sigma \to \tau \qquad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

The typed λ-calculus: semantics in **Set**

Types

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

Basic Types

$$[\![b]\!] = \text{(as given)}$$

Function spaces

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

Environments

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

Semantics: very explicit

Terms

$$\frac{\Gamma \vdash M : \sigma}{[\![M]\!] : [\![\Gamma]\!] \to [\![\sigma]\!]}$$

Variables

$$[\![x_i]\!](a_1, \ldots, a_n) = a_i$$

Constants

$$[\![c]\!](a_1, \ldots, a_n) = (\text{as given})$$

Abstractions

$$[\![\lambda x : \sigma.\, M]\!](a_1, \ldots, a_n) = a \in [\![\sigma]\!] \mapsto [\![M]\!](a_1, \ldots, a_n, a)$$

Applications

$$[\![MN]\!](a_1, \ldots, a_n) = [\![M]\!](a_1, \ldots, a_n)([\![N]\!](a_1, \ldots, a_n))$$

Semantics of Exceptions

Suppose programs can raise exceptions $e \in E$. Then we want:

$$\frac{\Gamma \vdash M : \sigma}{[\![M]\!] : [\![\Gamma]\!] \to ([\![\sigma]\!] + E)}$$

(Remember: $X + Y = (\{0\} \times X) \cup (\{1\} \times Y)$)

Function spaces

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

Variables

$$[\![x_i]\!](a_1, \ldots, a_n) = \mathrm{inl}(a_i) = \eta(a_i)$$

Constants

$$[\![c]\!](a_1, \ldots, a_n) = \mathrm{inl}(\text{as given})$$

Semantics of Exceptions (cntnd.)

Abstractions

$$[\![\lambda x : \sigma.\, M]\!](a_1, \ldots, a_n) = \eta(a \in [\![\sigma]\!] \mapsto [\![M]\!](a_1, \ldots, a_n, a))$$

Applications

For $M : \sigma \to \tau$, $N : \sigma$

$$[\![MN]\!](\gamma) = \text{E-ap}([\![M]\!](\gamma), [\![N]\!](\gamma))$$

where

$$\text{E-ap} : ([\![\sigma \to \tau]\!] + E) \times ([\![\sigma]\!] + E) \longrightarrow ([\![\tau]\!] + E)$$

$$\text{E-ap}(a, b) = \begin{cases} \text{inr}(e) & (\text{if } a = \text{inr}(e)) \\ \text{inr}(e') & (\text{if } a = \text{inl}(f) \text{ and } b = \text{inr}(e')) \\ f(c) & (\text{if } a = \text{inl}(f) \text{ and } b = \text{inr}(c)) \end{cases}$$

Moggi's insight: a categorical view of - + E

Functorial action

$$\frac{f : X \to Y}{f + E : X + E \to Y + E}$$

where:

$$(f + E)(a) = \begin{cases} \mathrm{inl}(f(b)) & (\text{if } a = \mathrm{inl}(b)) \\ \mathrm{inr}(e) & (\text{if } a = \mathrm{inr}(e)) \end{cases}$$

Monadic structure

Unit $\eta : X \to X + E$

Multiplication $\mu : (X + E) + E \to X + E$

$$\mu(a) = \begin{cases} \mathrm{inr}(e) & (\text{if } a = \mathrm{inr}(e)) \\ \mathrm{inr}(e') & (\text{if } a = \mathrm{inl}(\mathrm{inr}(e))) \\ \mathrm{inl}(b) & (\text{if } a = \mathrm{inl}(\mathrm{inl}(b)) \end{cases}$$

A categorical view of - + E (cntnd.)

Strength

Left strength

$$\mathrm{lst} : X \times (Y + E) \longrightarrow (X \times Y) + E$$

Right strength

$$\mathrm{rst} : (X + E) \times Y \longrightarrow (X \times Y) + E$$

$$\mathrm{lst}(\langle a, b \rangle) = \begin{cases} \mathrm{inr}(e) & (\text{if } b = \mathrm{inr}(e)) \\ \mathrm{inl}(\langle a, c \rangle) & (\text{if } b = \mathrm{inl}(c)) \end{cases}$$

Putting these together

$$
\begin{array}{lll}
((X \Rightarrow (Y + E)) + E) \times (X + E) & \xrightarrow{\text{rst}} & ((X \Rightarrow (Y + E)) \times (X + E)) + E \\
& \xrightarrow{\text{lst}+E} & ((X \Rightarrow (Y + E)) \times X) + E) + E \\
& \xrightarrow{\mu} & ((X \Rightarrow (Y + E)) \times X) + E \\
& \xrightarrow{\text{ap}+\text{E}} & (Y + E) + E \\
& \xrightarrow{\mu} & Y + E
\end{array}
$$

Strong monads

Summarising, one needs a operator $T(X)$ on **Set**, the category of sets, equipped with:

- A functorial action $(X \Rightarrow Y) \xrightarrow{T(\,\cdot\,)} (T(X) \Rightarrow T(Y))$

This makes T a functor

- A unit $X \xrightarrow{\eta X} T(X)$
- A multiplication $T(T(X)) \xrightarrow{\mu X} T(X)$

These make T a monad

- A (left) strength $X \times T(Y) \xrightarrow{\mathrm{st}_{X,Y}} T(X \times Y)$

This makes T a strong monad

Note, can derive the right strength:

$$T(X) \times Y \xrightarrow{twist} Y \times T(X) \xrightarrow{\mathrm{st}_{Y,X}} T(Y \times X) \xrightarrow{T(\mathrm{twist})} T(X \times Y)$$

Semantics of application for any monad

$$
\begin{array}{lll}
T(X \Rightarrow T(Y)) \times T(X) & \xrightarrow{\text{rst}} & T((X \Rightarrow T(Y)) \times T(X)) \\
& \xrightarrow{T(\text{lst})} & T(T((X \Rightarrow T(Y)) \times X)) \\
& \xrightarrow{\mu} & T((X \Rightarrow T(Y)) \times X) \\
& \xrightarrow{T(\text{ap})} & T(T(Y)) \\
& \xrightarrow{\mu} & T(Y)
\end{array}
$$

Moggi's insight (cntnd.)

Other effects can also be modelled by strong monads T, e.g.:

- State: functions $S \times X \xrightarrow{f} S \times Y$ can be rewritten as $X \xrightarrow{g} (S \times Y)^S$, and $T_{\text{state}}(X) = (S \times X)^S$ is a strong monad.
- Finite Nondeterminism: $T_{\text{SL}}(X) = \mathcal{F}^+(X)$ the collection of non-empty finite subsets of X.
- Continuations: Functions $R^Y \xrightarrow{f} R^X$ can be rewritten as $X \xrightarrow{g} T_{\text{cont}}(Y)$, where $T_{\text{cont}}(X) = (X \Rightarrow R) \Rightarrow R$.
- Selection: $T_{\text{sel}}(X) = (X \Rightarrow R) \Rightarrow X$ (Escardó and Oliva).

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

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

In **Cpo** one has similar examples. They include lifting to model recursion, e.g., for state + nontermination: $T(P) = ((S \times P)_\perp)^S$

Outline

Outline

Two questions

- How do effects arise, i.e., how do we "construct" them in a programming language?
- Answering that leads to understanding where (most of) Moggi's monads come from.

An example: finite nondeterminism

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* to the language:

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

with semantics

$$[\![M + N]\!](\gamma) = [\![M]\!](\gamma) \cup [\![N]\!](\gamma)$$

So $T_{\mathrm{SL}}(X)$ is an algebra (when equipped) with the binary operation $\cup : T_{\mathrm{SL}}(X) \times T_{\mathrm{SL}}(X) \to T_{\mathrm{SL}}(X)$. But which algebra is it?

Nondeterminism as an algebraic effect

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

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

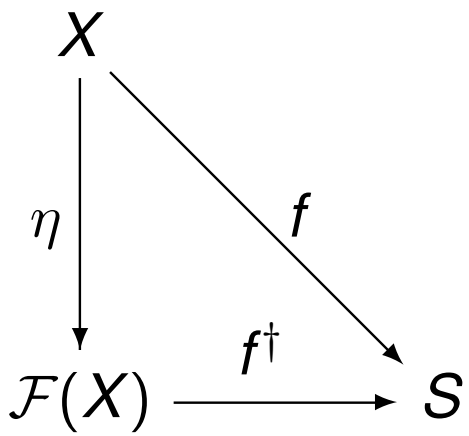
The above algebra on $\mathcal{F}^+(X)$ satisfies these equations, interpreting $+$ as $\cup$.

Further:

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

Freeness

For any semilattice S, and any function $f : X \to S$ there is a unique homomorphism $f^\dagger : \mathcal{F}^+(X) \to S$ such that the following diagram commutes:



where

1. $\eta(x) = \{x\}$
2. $f^\dagger(\{x_1, \ldots, x_n\}) = f(x_1) + \ldots + f(x_n)$

when: $\mathcal{F}(X)$ is a strong monad with unit η, multiplication $(\mathrm{id}_{\mathcal{F}(X)})^\dagger$.

Is this the right set of axioms for nondeterminism?

- 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

SL *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. □

Other effects

- Similar results hold in **Set** for, eg, exceptions, (global) state; I/O; (probabilistic) nondeterminism; and combinations thereof (but may not get HP completeness).
- May need infinitary algebra and parameterised operations.

Further

- Works similarly for **Cpo** but also need inequations $t \leq u$.
- For locality, e.g., new variables and fresh names, one uses categories of presheaves.

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; this will prove useful.
- Equations $t = u$
- Axiomatisations Sets Ax of equations
- Deduction $\mathrm{Ax} \vdash t = u$
- Theories Sets of equations Th closed under equational deduction

Finitary equational theories: equational deduction rules

Axiom

$$\mathrm{Ax} \vdash t = u \quad (\text{if } t = u \in \mathrm{Ax})$$

Equivalence

$$\mathrm{Ax} \vdash t = t \qquad \frac{\mathrm{Ax} \vdash t = u \quad \mathrm{Ax} \vdash u = v}{\mathrm{Ax} \vdash u = v} \qquad \frac{\mathrm{Ax} \vdash t = u}{\mathrm{Ax} \vdash u = t}$$

Congruence

$$\frac{\mathrm{Ax} \vdash t_i = u_i \quad (i = 1, n)}{\mathrm{Ax} \vdash f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)} \quad (\text{for } f : n)$$

Substitution

$$\frac{\mathrm{Ax} \vdash t = u}{\mathrm{Ax} \vdash t[v/x] = u[v/x]}$$

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 $\rho : \mathrm{Var} \to A$.

$$\mathcal{A}[\![x]\!](\rho) = \rho(x) \qquad \mathcal{A}[\![\mathrm{op}(t_1, \ldots, t_n)]\!](\rho) = \mathrm{op}_{\mathcal{A}}(\mathcal{A}[\![t_1]\!](\rho), \ldots, \mathcal{A}[\![t_n]\!](\rho))$$

- 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 is the algebra with carrier:

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

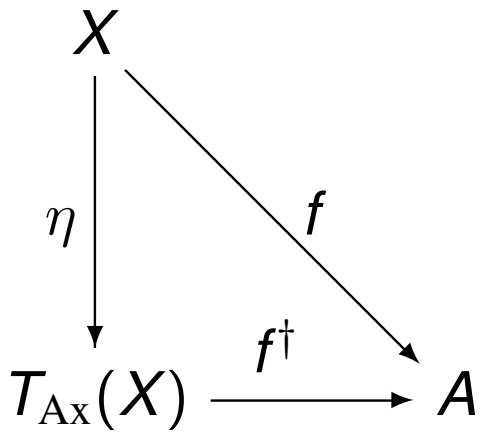
where $[t]_{\mathrm{Ax}} =_{\mathrm{def}} \{u \mid \mathrm{Ax} \vdash t = u\}$.

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:



where:

1. $\eta =_{\mathrm{def}} x \mapsto [x]_{\mathrm{Ax}}$
2. $f^\dagger([t]) = \mathcal{A}[\![t]\!](f)$

when: $T_{\mathrm{Ax}}(X)$ is a strong monad with unit η, multiplication $(\mathrm{id}_{T_{\mathrm{Ax}}(X)})^\dagger$. Above only characterises F_{Ax} up to (algebraic) isomorphism (e.g., consider SL).

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: one boolean location

Signature: $\mathrm{write}_{\mathrm{true}}, \mathrm{write}_{\mathrm{false}} : 1; \mathrm{read} : 2$.

- Read $\mathrm{write}_b(x)$ as "write b, and continue with x"
- Read $\mathrm{read}(x, y)$ as "read the boolean variable, continuing with x, if true, and with y otherwise".

Axioms The set of axioms $\mathrm{BoolState}$

$$\mathrm{read}(x, x) = x$$

$$\mathrm{read}(\mathrm{read}(w, x), \mathrm{read}(y, z)) = \mathrm{read}(w, z)$$

$$\mathrm{write}_b(\mathrm{write}_{b'}(x)) = \mathrm{write}_{b'}(x)$$

$$\mathrm{read}(\mathrm{write}_{\mathrm{true}}(x), \mathrm{write}_{\mathrm{false}}(y)) = \mathrm{read}(x, y)$$

$$\mathrm{write}_{\mathrm{true}}(\mathrm{read}(x, y)) = \mathrm{write}_{\mathrm{true}}(x)$$

$$\mathrm{write}_{\mathrm{false}}(\mathrm{read}(x, y)) = \mathrm{write}_{\mathrm{false}}(y)$$

One boolean location (cntnd)

The monad is:

$$T_{\mathrm{bool}}(X) = (\mathbb{T} \times X)^{\mathbb{T}}$$

The unit is $\eta(x)(b) = (b, x)$

Yet another example: probabilistic computation

Signature: binary operations $+_p$ for $p \in [0,1]$.
Read $+_p$ as 'do x with probability p and y with probability $1-p$'

Axioms: the barycentric algebra axioms:

One	$x +_1 y = x$
ID	$x +_r x = x$
SC	$x +_r y = y +_{1-r} x$
SA	$(x +_p y) +_r z = x +_{pr} (y +_{\frac{r(1-p)}{1-pr}} z) \quad (r < 1, p < 1)$

A consequence:

COM $\quad (x +_r u) +_s (v +_r y) = (x +_s v) +_r (u +_s y)$

Yet another example: probabilistic computation

The monad is the set of finite probability distributions over X

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

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

Remarks

- $\mathbb{P}\mathrm{rob}$ is not HP complete, but its only proper equational extension is (equivalent to) $\mathbb{SL}$, the theory of semilattices (this is a non-trivial result). These have an associative, commutative, idempotent binary operator.
- $\mathbb{P}\mathrm{rob}$ can be alternatively axiomatised using n-ary operations $\sum_{n,p_1,\ldots,p_n}$ for all $n \geq 0$ and n-tuples of nonnegative reals $p_1, \ldots, p_n$ summing to 1. The axioms are:
  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 Kronecker's δ^i_j is 1 if $i = j$, and 0 otherwise.

Algebraic semantics

Form of semantics

$$\frac{\Gamma \vdash M : \sigma}{[\![\Gamma]\!] \xrightarrow{[\![M]\!]} T_{\mathrm{Ax}}([\![\sigma]\!])}$$

Abstraction: Typing

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma.\, M : \sigma \to \tau}$$

Abstraction: Semantics

$$[\![\lambda x : \sigma.\, M]\!](\gamma) = [a \in [\![\sigma]\!] \mapsto [\![M]\!](\gamma, a)]_{Ax}$$

Algebraic semantics (cntnd.)

Application: Typing

$$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

Application: Semantics

Suppose

$$[\![M]\!](\gamma) = [t(f_1, \ldots, f_m)]_{\mathrm{Ax}}$$
$$[\![N]\!](\gamma) = [u(a_1, \ldots, a_n)]_{\mathrm{Ax}}$$
$$f_i(a_j) = [v_{ij}]_{\mathrm{Ax}}$$

Then

$$[\![MN]\!](\gamma) = [t(u(v_{11}, \ldots, v_{1n}), \ldots, u(v_{m1}, \ldots, v_{mn}))]_{\mathrm{Ax}}$$

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:

$$\begin{array}{ccc} T_{\mathrm{Ax}}(X)^n & \xrightarrow{\mathrm{op}_{F_{\mathrm{Ax}}(X)}} & T_{\mathrm{Ax}}(X) \\ \downarrow{\scriptstyle (f^\dagger)^n} & = & \downarrow{\scriptstyle f^\dagger} \\ T_{\mathrm{Ax}}(Y)^n & \xrightarrow[\mathrm{op}_{F_{\mathrm{Ax}}(Y)}]{} & T_{\mathrm{Ax}}(Y) \end{array}$$

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

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, with $\mathcal{E}$ alternatively of the forms $[\cdot]N$ or $(\lambda x : \sigma.\, N)[\cdot]$

$$\models (\mathrm{raise}_e())(N) = \mathrm{raise}_e() \qquad \models (\lambda x : \sigma.\, M)\mathrm{raise}_e() = \mathrm{raise}_e()$$

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

$$\models (\lambda x : \sigma.\, M)(N +_p N') = (\lambda x : \sigma.\, M)N +_p (\lambda x : \sigma.\, M)N'$$

Generic effects

- Given an algebraic family $T_{\mathrm{Ax}}(X)^n \xrightarrow{\varphi X} T_{\mathrm{Ax}}(X)$, regarding n as $\{0, \ldots, n-1\}$, and setting $X = n$, 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:

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

- This correspondence is a bijection between algebraic families and generic effects.
- Noting that $T_{\mathrm{Ax}}(n)$ 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 $0 + 1$.

- 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 $0 +_p 1$.

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

Outline

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 get expected op. sems., eg, state. (P. & Power; Kammar et al)
- 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. (P. & Pretnar; Bauer & Pretnar; Kammar, Oury & Lindley)
- Combining monads in terms of combining theories, primarily sum and tensor. (Hyland,P. & Power)
- Work on combining algebraic effects with continuations, which are not algebraic and require special treatment. (Hyland, Levy, Power & P.)
- First thoughts on a general logic of effects; connects with modal logic. Does not give Hoare logic. (P. & Pretnar)
- ✓ Type and effect systems. (Kammar & P.; Katsumata; Bauer & Pretnar)
- Work on locality and effects (Staton; Melliès; P. & Power; Power).
- Algebraic accounts of control (Fiore & Staton)

Exercise 1: An extension of the typed λ-calculus

Consider the following extension of the λ-calculus.
Raw Syntax

Types $\quad \sigma ::= \mathrm{bool} \mid \mathrm{unit} \mid \sigma \times \tau \mid \sigma \to \tau$

Terms $\quad M ::= \mathrm{true} \mid \mathrm{false} \mid \mathrm{if}\, A\, \mathrm{then}\, M\, \mathrm{else}\, N \mid x \mid$
$* \mid (M, N) \mid \mathrm{fst}(M) \mid \mathrm{snd}(M) \mid$
$\lambda x : \sigma.\, M \mid MN \mid$
$\mathrm{op}(M_1, \ldots, M_n)$

Informal understanding of the language.

- bool has two values true and false. The conditional uses them.
- unit has one value $*$.
- $\sigma \times \tau$ consists of pairs of elements of σ and τ, given by (M, N) and accessed by fst and snd

The Exercise

- Write down typing rules for this language
- Give it a monadic semantics
- Give it an algebraic semantics
- Define evaluation contexts for this language.
- Prove that evaluation contexts commute with operations. (That is, show the equality given above for the relation between evaluation contexts and operations holds in the semantics.)

Exercise 2: Looking at monads

Recall the claimed example monads covered in the lecture:

State

$$T_{\text{state}}(X) = (S \times X)^S \qquad T_{\text{bool}}(X) = (\mathbb{T} \times X)^{\mathbb{T}}$$

Finite Nondeterminism

$$T_{\text{SL}}(X) = \mathcal{F}^+(X)$$

Continuations

$$T_{\text{cont}}(X) = R^{R^X}$$

State plus exceptions

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

Finite Probabilistic Nondeterminism

$$T_{\text{prob}}(X) = \mathcal{D}_\omega(X)$$

Exercise

1. Choose a monad or two (I recommend state) and find
   - their functorial actions
   - their unit and multiplication,
   - their (left) strength
   - the algebraic structure of $T(X)$ (for example $+ : T_{\mathrm{SL}}(X)^2 \to T_{\mathrm{SL}}(X)$ is union).

2. (More advanced)
   - Show that every monad on **Set** has a unique (left or right) strength.
   - Show that T_{cont} is not given by any algebraic theory.

Exercise 3: Free Algebra Monads

Show that T_{Ax} is indeed a strong monad, i.e.:

- Find its functorial action.
- Find its unit (well that was given) and its multiplication
- Find its strength (Hint: if A is an algebra then A^X becomes one too).

Exercise 4: Identifying monads

It is claimed above that some monads are the free algebra monads associated to certain equational theories. Prove this!

Method We know the monads are given, up to isomorphism, by the equational logic as free algebra monads. So:

- Associate to every term t a normal form $|t|$ so that $\vdash_{\mathrm{Ax}} t = u$ iff $|t| = |u|$.
- Inspect the normal forms to see how they correspond to elements of $T(\mathrm{Var})$.
- For example, for semilattices, every term can be proved equal to one of the form $x_1 + ... + x_n$ (ignoring brackets) with no x_i repeated, and the x_i taken in order according to some fixed enumeration of the variables.
- (If you are trying probability, use the alternative theory.)

Exercise 5: HP completeness

- Show that $\mathbb{B}\mathrm{ool}\mathbb{S}\mathrm{tate}$ is Hilbert Post complete. (Hint: see what happens if you assume two different normal forms equal).
- The axiomatisation $\mathbb{B}\mathrm{ool}\mathbb{S}\mathrm{tate}$ is redundant. That you can derive some of the axioms from the others. Which ones are they?

Exercise 6: Probability

- Show the two axiomatisations are equivalent. That is, one can define each in terms of the other. More precisely:
  - In one direction, define $x +_p y$ to be $px + (1 - p)y$ then, using the alternative axiomatisation, show that all the barycentric axioms are derivable.
  - In the other direction define the operations $\sum_{n,p_1,\ldots,p_n}$ in terms of the barycentric operations, and show all the axioms of the alternative theory are derivable.
  - Then show the compositions of the two translations are provably equal in the relevant theories to the identity translation.
    (The definition of translations between equational theories is in the next lecture.)

Probability (cntnd)

- Prove Neumann's Lemma, which is that if you add one equation of the form $x +_p y = x +_q y$ with $0 < p < q < 1$ to Prob then one can derive all such equations (this is not so easy).
- Show that the only nontrivial extension of Prob is SL the theory of semilattices.
  Method: show that if you add one equation between two different normal forms then the resulting theory is either equationally inconsistent or else it is intertranslatable with SL; you will need Neumann's lemma.