Lecture 3: Algebraic Effects II

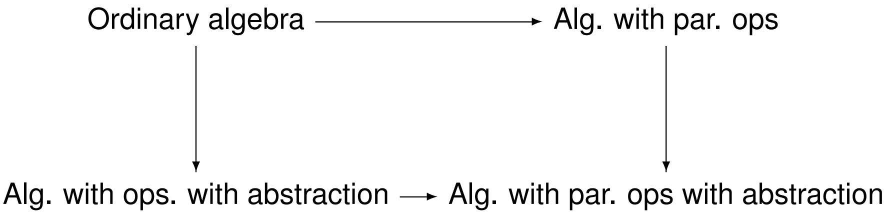
Gordon Plotkin

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

Outline

Plan of Lecture



Outline

Parametric finitary equational theories: syntax

- First-order multi-sorted signature

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

- Parametric signature

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

- 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 $\mathrm{interval}$, constants $0, 1$, binary function symbols $\times$, a unary function symbol $1-$, and a relation symbol $<$.
  Σ_e has a single binary operation symbol $+ : \mathrm{interval}; 2$.
  Here is an example equation:

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

Addition to λ-calculus syntax

- Types

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

- Terms

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

- 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

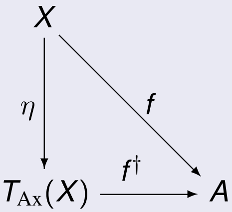
$$\mathcal{A}[\![\mathrm{op}_{u_1,\ldots,u_m}(t_1, \ldots, t_n)]\!](\rho_p, \rho_e) = \\ \mathrm{op}_{\mathcal{A}}(\mathcal{M}[\![u_1, \ldots, u_m]\!](\rho_p), \mathcal{A}[\![t_1]\!](\rho_p, \rho_e), \ldots, \mathcal{A}[\![t_n]\!](\rho_p, \rho_e))$$

  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 an $\eta : X \to T_{\mathrm{Ax}}(X)$, 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:



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.

Outline

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 the 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

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

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

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

$$\mathrm{lookup}_{l}(m : \mathrm{nat}.\, \mathrm{lookup}_{l'}(n : \mathrm{nat}.\, x(m,n))) = \mathrm{lookup}_{l'}(n : \mathrm{nat}.\, \mathrm{lookup}_{l}(m : \mathrm{nat}.\, x(m,n))) \quad (l \neq l') \tag{8}$$

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

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

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

  with a subcollection $\mathrm{Ar} \subseteq S$ of *arity* sorts

- Parametric signature

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

- Terms

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

- 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 $s \in \mathrm{Ar}$.
- 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.

An Example: State

- 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. The rest, including $\mathcal{M}$, is as suits the purpose at hand.
- Operation symbols $\mathrm{input} : \varepsilon; \mathrm{in}$ and $\mathrm{output} : \mathrm{out}; 1$
- Algebraic 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.

Outline

Algebraic operations, somewhat more generally

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

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

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

$$\begin{array}{ccc}
\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]} & \xrightarrow{\mathrm{op}_{F_{\mathrm{Ax}}(X)}} & T_{\mathrm{Ax}}(X) \\
{\scriptstyle \mathrm{id}_{\mathcal{M}[\![\mathbf{s}]\!]} \times (f^\dagger)^{\mathcal{M}[\![\mathbf{s}_1]\!]}} \downarrow & = & \downarrow {\scriptstyle f^\dagger} \\
\mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(Y)^{\mathcal{M}[\![\mathbf{s}_1]\!]} & \xrightarrow[\mathrm{op}_{F_{\mathrm{Ax}}(Y)}]{} & T_{\mathrm{Ax}}(Y)
\end{array}$$

We again call such a family of functions φ_X algebraic.

Generic effects, somewhat more generally

- Given an algebraic family

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

  we obtain the generic effect:

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

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

$$\begin{array}{lll} \mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{\mathcal{M}[\![\mathbf{s}_1]\!]} & \xrightarrow{\mathrm{id}_{\mathcal{M}[\![\mathbf{s}]\!]} \times (\,\cdot\,)^{\dagger}} & \mathcal{M}[\![\mathbf{s}]\!] \times T_{\mathrm{Ax}}(X)^{T_{\mathrm{Ax}}(\mathcal{M}[\![\mathbf{s}_1]\!])} \\ & \xrightarrow{e \times \mathrm{id}} & T_{\mathrm{Ax}}(\mathcal{M}[\![\mathbf{s}_1]\!]) \times T_{\mathrm{Ax}}(X)^{T_{\mathrm{Ax}}(\mathcal{M}[\![\mathbf{s}_1]\!])} \\ & \xrightarrow{\mathrm{ev}} & T_{\mathrm{Ax}}(X) \end{array}$$

- 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

$$\mathrm{Loc} \times \mathbb{N} \times T_S(X)^{\mathbb{1}} \xrightarrow{\mathrm{update}_{F_S(X)}} T_S(X)$$

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

$$\mathrm{write}(d) = \mathrm{in}_2(d, \mathrm{in}_3(*))$$

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