

## Lecture 3: Algebraic Effects II

Gordon Plotkin

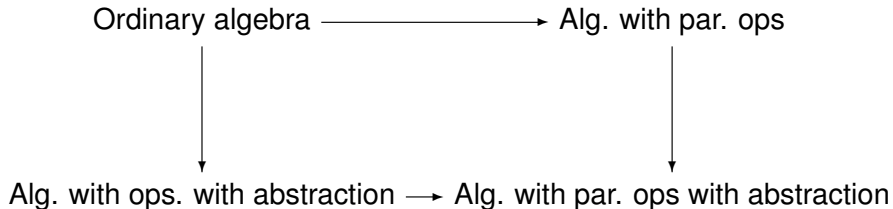
Laboratory for the Foundations of Computer Science, School of Informatics,  
University of Edinburgh

Beihang University  
October, 2015

# Outline

- 1 Algebra with parameterised operations
- 2 Algebra with parameters and parametric arguments
- 3 Algebraic operations and generic effects (cntnd.)

# Plan of Lecture



# Outline

- 1 Algebra with parameterised operations
- 2 Algebra with parameters and parametric arguments
- 3 Algebraic operations and generic effects (cntnd.)

# Parametric finitary equational theories: syntax

- **First-order multi-sorted signature**

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

- **Parametric signature**

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

- **Terms**

$$t ::= x \quad | \quad \text{op}_{u_1, \dots, u_m}(t_1, \dots, t_n) \quad (\text{op} : s_1, \dots, s_m; n \text{ and } u_i : s_i).$$

- **Equations**  $t = u$  ( $\varphi$ ) where  $\varphi$  is a first-order formula over  $\Sigma_p$ .
- **Axiomatisations** Sets  $A_x$  of equations
- **Deduction** (an interesting question, not treated here)

# Examples

- Exceptions**  $\Sigma_p$  has a single sort  $exc$ , and constants  $e : 0$  for each  $e \in E$ .  
 $\Sigma_e$  has a single operation symbol  $raise : exc; 0$ . There are no equations.
- Probability**  $\Sigma_p$  has a single sort  $interval$ , constants  $0, 1$ , binary function symbols  $\times$ , a unary function symbol  $1-$ , and a relation symbol  $<$ .  
 $\Sigma_e$  has a single binary operation symbol  $+ : interval; 2$ .  
 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 $\lambda$ -calculus syntax

- Types

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

- Terms

$$M ::= f(M_1, \dots, M_n) \ (f \in \text{Fun}) \mid P(M_1, \dots, M_n) \ (P \in \text{Pred}) \mid \\ \text{true} \mid \text{false} \mid \text{if } L \text{ then } M \text{ else } N \mid \\ \text{op}_{M_1, \dots, M_m}(N_1, \dots, N_n)$$

- Example type-checking rule

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

where  $\text{op} : s_1, \dots, s_m; n$

# Parametric finitary equational theories: semantics

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

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

where  $\mathcal{M}[\mathbf{s}_1, \dots, \mathbf{s}_m] =_{\text{def}} \mathcal{M}[\mathbf{s}_1] \times \dots \times \mathcal{M}[\mathbf{s}_m]$ .

**Homomorphisms** are then defined in the evident way.

- **Denotation**  $\mathcal{A}[\mathbf{t}](\rho_p, \rho_e)$ , where  $\rho_e : \text{Var} \rightarrow A$ . For example

$$\begin{aligned} & \mathcal{A}[\text{op}_{u_1, \dots, u_m}(t_1, \dots, t_n)](\rho_p, \rho_e) = \\ & \text{op}_{\mathcal{A}}(\mathcal{M}[u_1, \dots, u_m](\rho_p), \mathcal{A}[t_1](\rho_p, \rho_e), \dots, \mathcal{A}[t_n](\rho_p, \rho_e)) \end{aligned}$$

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



# Free algebra theorem

## Theorem

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

$$\begin{array}{ccc} X & & \\ \eta \downarrow & \searrow f & \\ T_{A_X}(X) & \xrightarrow{f^\dagger} & \mathcal{A} \end{array}$$

# Idea of proof

The idea is to reduce to ordinary equational theories.

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

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

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

# Outline

- 1 Algebra with parameterised operations
- 2 Algebra with parameters and parametric arguments
- 3 Algebraic operations and generic effects (cntnd.)

# State treated algebraically

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

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

But

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

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:

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

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:

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

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

## Infinitary equational theories: semantics

**Algebras** are structures  $\mathcal{A} = (A, \text{op}_{\mathcal{A}} : A^n \rightarrow A \text{ (op : } n))$ , and recall that here  $n$  can be  $\omega$ .

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

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

**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 $T_{A_X}$ of an infinitary axiomatic theory $A_X$

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

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

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

$$\text{op}_{F_{A_X}(X)}([t]) = [\text{op}(t)] \quad (\text{op} : n)$$

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

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

and the **multiplication** is  $(\text{id}_{T_{A_X}(X)})^\dagger$ .



## Notation and equations for state

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

Equations for writing and reading a single location:

$$\text{update}_{l,m}(\text{update}_{l,n}(x)) = \text{update}_{l,n}(x) \quad (1)$$

$$\text{lookup}_l(m : \text{nat. } \text{lookup}_l(n : \text{nat. } x(m, n))) = \text{lookup}_l(m : \text{nat. } x(m, m)) \quad (2)$$

$$\text{lookup}_l(n : \text{nat. } x) = x \quad (3)$$

$$\text{update}_{l,m}(\text{lookup}_l(n : \text{nat. } x(n))) = \text{update}_{l,m}(x(m)) \quad (4)$$

$$\text{lookup}_l(n : \text{nat. } \text{update}_{l,n}(x)) = x \quad (5)$$

## Notation and equations for state (cntnd)

### Commutation Equations for different locations

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

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

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

# Redundancies

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

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

## Parametric axiom. ths. with abstraction: syntax

- **First-order multi-sorted signature**

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

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

- **Parametric signature**

$$\Sigma_e = (\text{Op}, \text{ar}_{\text{op}} : \text{Op} \rightarrow \mathcal{S}^* \times \text{Ar}^{**})$$

- **Terms**

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

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

# Addition to $\lambda$ -calculus syntax

- Types

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

- Terms

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

- Example type-checking rule

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

where  $\text{op} : \mathbf{s}; \mathbf{s}_1, \dots, \mathbf{s}_m$

## Parametric axiom. ths. with abstraction: semantics

- **Parameter interpretation** We fix an interpretation  $\mathcal{M}$  of  $\Sigma_p$ , such that  $\mathcal{M}[\mathbf{s}]$  is countable for all  $\mathbf{s} \in \text{Ar}$ .
- **Algebras** With that, a  $\Sigma_e$ -algebra is a structure

$$(A, \text{op}_{\mathcal{A}} : \mathcal{M}[\mathbf{s}] \times A^{\mathcal{M}[\mathbf{s}_1]} \times \dots \times A^{\mathcal{M}[\mathbf{s}_n]} \rightarrow A \quad (\text{op} : \mathbf{s}; \mathbf{s}_1, \dots, \mathbf{s}_n))$$

- **Denotation**  $\mathcal{A}[\mathbf{t}](\rho_p, \rho_e)$ , where  $\rho_e : \text{Var} \rightarrow A$ . For example

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

where:

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

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

# Free algebras, etc.

- As usual, there is a free algebra  $F_{Ax}(X)$  over any set  $X$ , which induces the corresponding monad  $T_{Ax}(X)$ .
- 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  $\text{loc}, \text{nat}$ , and there is a predicate symbol  $=: \text{loc}, \text{loc}$ . We assume  $\mathcal{M}[\text{=}]$  is equality,  $\mathcal{M}[\text{loc}]$  is finite, and  $\mathcal{M}[\text{nat}] = \mathbb{N}$ . Set  $\text{Loc} =_{\text{def}} \mathcal{M}[\text{loc}]$ .
- **Axioms**  $\text{Ax}_S$  is as above.
- **Monad**  $T_S(X) = (S \times X)^S$ , where  $S =_{\text{def}} \mathbb{N}^{\text{Loc}}$
- **Operations**

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

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

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

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



## 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** input :  $\varepsilon$ ; in and output : out; 1
- **Algebraic Axioms** None!
- **Monad**  $T_{I/O}(X)$  is the least set  $Y$  such that:

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

and we just write:

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

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

## I/O cntnd.

- Operations

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

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

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

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

## The general case, when there are no axioms

We have:

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

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

# Outline

- 1 Algebra with parameterised operations
- 2 Algebra with parameters and parametric arguments
- 3 Algebraic operations and generic effects (cntnd.)

## Algebraic operations, somewhat more generally

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

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

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

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

We again call such a family of functions  $\varphi_X$  **algebraic**.

# Generic effects, somewhat more generally

- Given an algebraic family

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

we obtain the **generic effect**:

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

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

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

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

## An example: side-effects

- **Lookup** The generic effect corresponding to

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

is

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

where

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

- **Update** The generic effect corresponding to

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

is

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

where

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

## Another example: interactive I/O

- **Input** The generic effect corresponding to

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

is

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

where

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

- **Output** The generic effect corresponding to

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

is

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

where

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



## 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  $op : \mathbf{s}; \mathbf{s}_1, \dots, \mathbf{s}_m$  we have:

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

assuming variable clashes are avoided.