Semantics of Computational Effects and Effect Systems

Shin-ya Katsumata

National Institute of Informatics

Shonan school
15 May, 2017

Plan of the Lecture

- Monads and computational effects
- Relating monadic semantics
- Effect systems
- Effect soundness
- Graded monads and related categorical structures

Part I

Computational Effects and Monads

Monads

1. Monads in a Category

Any endofunctor $T: X \to X$ has composites $T^2 = T \circ T: X \to X$ and $T^3 = T^2 \circ T: X \to X$. If $\mu: T^2 \dot\to T$ is a natural transformation, with components $\mu_x: T^2 x \to Tx$ for each $x \in X$, then $T\mu: T^3 \dot\to T^2$ denotes the natural transformation with components $(T\mu)_x = T(\mu_x): T^3 x \to T^2 x$ while $\mu T: T^3 \dot\to T^2$ has components $(\mu T)_x = \mu_{Tx}$. Indeed, $T\mu$ and μT are "horizontal" composites in the sense of § II.5.

Definition. *A monad $T = \langle T, \eta, \mu \rangle$ in a category X consists of a functor $T: X \to X$ and two natural transformations*

$$\eta: I_X \dot\to T, \qquad \mu: T^2 \dot\to T \tag{1}$$

which make the following diagrams commute

$$\begin{array}{ccc} T^3 & \xrightarrow{T\mu} & T^2 \\ {\scriptstyle \mu T}\downarrow & & \downarrow{\scriptstyle \mu} \\ T^2 & \xrightarrow{\mu} & T, \end{array} \qquad \begin{array}{ccccc} IT & \xrightarrow{\eta T} & T^2 & \xleftarrow{T\eta} & TI \\ \| & & \downarrow{\scriptstyle \mu} & & \| \\ T & = & T & = & T. \end{array} \tag{2}$$

Partial photocopy of p. 137 of Saunders Mac Lane.
Categories for the Working Mathematician (2nd ed).
Springer, 1998.

Monads

- The structure was descerned by Godement (standard construction / triple).
  - R. Godemant. Topologie Algébrique et Théorie des Faisceaux. Hermann 1958.
- Jean Bénabou coined the word **monad** in 1966.

  Michael Barr. Subject: Re: Where does the term monad come from?
  Newsgroups: gmane.science.mathematics.categories, Wednesday 1st April 2009 18:13:55 UTC
  http://permalink.gmane.org/gmane.science.mathematics.categories/214
- Moggi applied it to represent the notions of computation:
  - E. Moggi. Computational Lambda-Calculus and Monads. In Proc. LICS, 1989.

    $\Longrightarrow$ λ_c-calculus
  - E. Moggi. Notions of computation and monads. Information and Computation 93 (1), 1991

    $\Longrightarrow$ λ_{ML}-calculus

Monads

... Then, in 1987, Eugenio Moggi completed his PhD thesis at the University of Edinburgh under Gordon Plotkin, with Martin Hyland his external examiner. At precisely that point, Moggi's new idea of computational effects came to the attention of experienced category theorists.

A defining moment came at Moggi's oral defence. Moggi had completed a technical thesis on partiality, and the discussion turned to future work. He then introduced his new idea of notions of computation and proposed using monads to model them. It immediately struck Hyland as a particularly elegant idea, involving an enrichment of a basic type theory with terms having computational meaning. He was very encouraging. ...

Quoted from p. 451 of M. Hyland and J. Power.
The Category Theoretic Understanding of Universal Algebra: Lawvere Theories and Monads
ENTCS 172, April, 2007, pp. 437–458

Monad (as Kleisli Triple) on **Set**

Definition

A **monad** (on **Set**) consists of:

- T **sending** a set A to a set TA.
- **unit** function $\eta_A : A \to TA$.
- **Kleisli extension**
  $(-)^{\#} : (A \Rightarrow TB) \to (TA \Rightarrow TB)$.

They satisfy, for all $f : A \to TB, g : B \to TC$,

$$\eta_A^{\#} = \mathrm{id}_{A^*}, \quad f^{\#} \circ \eta_A = f, \quad (g^{\#} \circ f)^{\#} = g^{\#} \circ f^{\#}.$$

Following Haskell, define **x** $\gg=$ **f** to be $f^{\#}(x)$.

List Monad / Free Monoid Monad

- Kleene closure $(-)^*$.
- The unit function is

$$\eta_A : A \to A^*, \quad \eta_A(x) = (x)$$

- The Kleisli extension is

$$\frac{f : A \to B^*}{f^\# : A^* \to B^*}, \quad f^\#(a_1 \cdots a_n) = f(a_1) \cdots f(a_n)$$

Powerset Monad

- The powerset construction $\mathcal{P}$.
- The unit function is

$$\eta_A : A \to \mathcal{P}A, \quad \eta_A(x) = \{x\}$$

- The Kleisli extension is

$$\frac{f : A \to \mathcal{P}B}{f^{\#} : \mathcal{P}A \to \mathcal{P}B}, \quad f^{\#}X = \bigcup_{x \in X} f(x)$$

Monads from Algebraic Theory

- Σ: a ranked alphabet; e.g.

$$\Sigma = \{e^0, m^2\}$$

- E: a set of equational axioms on Σ-terms; e.g.

$$E = \{m(e, x) = x, m(x, e) = x,\\ m(x, m(y, z)) = m(m(x, y), z)\}$$

- $T_\Sigma A$: the set of Σ-terms over A

$$T_\Sigma\{1, 2, 3\} \ni m(1, m(e, 3))$$

- We call an E-equivalence class of $T_\Sigma A$ a **(Σ, E)-polynomial** over A.

Monads from Algebraic Theory

(Σ, E) determines a monad:

- TA = the set of (Σ, E)-polynomials over A.
- $\eta_A(a) = a$ (as a polynomial)
- For $f : A \to TB$, its extension $f^\# : TA \to TB$ performs the **simultaneous substitution**:

$$f^\#(t) = t[f(a)/a]_{a \in A}$$

Monads from Algebraic Theory

- $\eta_A^\# = \mathrm{id}_{TA}$ because

$$t[a/a]_{a \in A} = t$$

- $f^\# \circ \eta_A = f$ because

$$a[f(a)/a]_{a \in A} = f(a)$$

- $g^\# \circ f^\# = (g^\# \circ f)^\#$ because

$$t[f(a)/a]_{a \in A}[g(b)/b]_{b \in B} = t[f(a)[g(b)/b]_{b \in B}/a]_{a \in A}$$

Continuation Monad

- "Double negation" $\mathcal{C}^R X = (X \Rightarrow R) \Rightarrow R$
- The unit function is

$$\eta_A(a) = \lambda\rho\,.\,\rho a$$

- The Kleisli extension is

$$\frac{f : A \to \mathcal{C}^R B}{f^\# : \mathcal{C}^R A \to \mathcal{C}^R B}, \quad f^\# g = \lambda\rho\,.\,g(\lambda a\,.\,fa\rho)$$

Other Monads

- Writer monad $TA = \Delta^* \times A$
- State monad $TA = S \Rightarrow (A \times S)$
- Finite distribution monad
  $TA = \{f : A \to_{fin} [0,1] \mid \sum_{a \in A} f(a) = 1\}$
- ... and some combinations of them

Algebraic Operations [Plotkin&Power'03]

... is a family of functions:

$$\alpha_A : TA \times \cdots \times TA \to TA$$

$$\alpha_A(t_1, \cdots, t_n) \gg= f = \alpha_B(t_1 \gg= f, \cdots, t_n \gg= f),$$

corresponding to **derived operations** on polynomials.

$$(\cdot) : A^* \times A^* \to A^* \qquad (\cup) : \mathcal{P}A \times \mathcal{P}A \to \mathcal{P}A$$

$$p : A^* \times A^* \times A^* \to A^* \qquad p(x, y, z) = xyzyx$$

Exercise

Show that an n-ary algebraic operation for T bijectively corresponds to an element in $T\{1, \cdots, n\}$.

Why Monads in Semantics?

Let functions speak about what they do!

$$f : A \to B$$

Why Monads in Semantics?

Let functions speak about what they do!

$$f : A \to \mathbf{T}B$$

λ_c^{or}: Lambda Calculus with Choice

Extend the STLC with natural numbers:

$$\frac{}{\Gamma \vdash n : nat} \qquad \frac{\Gamma \vdash M : nat \quad \Gamma \vdash N : nat}{\Gamma \vdash M + N : nat}$$

and a choice operation:

$$\frac{\Gamma \vdash M_1 : \tau \quad \Gamma \vdash M_2 : \tau}{\Gamma \vdash M_1 \ \mathrm{or}\ M_2 : \tau}$$

Examples:

$$(3 \ \mathrm{or}\ 2) + (3 \ \mathrm{or}\ 2)$$
$$(\lambda x \,.\, x + x)(3 \ \mathrm{or}\ 2)$$
$$(\lambda x \,.\, x) \ \mathrm{or}\ (\lambda x \,.\, x + x)$$

Monadic Semantics of λ_c^{or}

Interpretation of types:

$$[\![nat]\!] = \mathbb{N}, \quad [\![\tau \Rightarrow \tau']\!] = [\![\tau]\!] \Rightarrow \mathbf{T}[\![\tau']\!]$$

Interpretation of judgements:

$$[\![M]\!] : [\![\Gamma]\!] \to \mathbf{T}[\![\tau]\!]$$

$$\begin{aligned}
[\![x]\!]\rho &= \eta(\rho(x)) \\
[\![\lambda x \,.\, M]\!]\rho &= \eta(\lambda v \,.\, [\![M]\!]\rho\{x \mapsto v\}) \\
[\![MN]\!]\rho &= [\![M]\!]\rho \gg\!= (\lambda m \,. \\
&\qquad [\![N]\!]\rho \gg\!= (\lambda n \,.\, m(n))) \\
[\![M \,\mathrm{or}\, N]\!]\rho &= \alpha([\![M]\!]\rho, [\![N]\!]\rho)
\end{aligned}$$

Monadic Semantics of λ_c^{or}

Interpretation of types:

$$[\![nat]\!] = \mathbb{N}, \quad [\![\tau \Rightarrow \tau']\!] = [\![\tau]\!] \Rightarrow \mathbf{T}[\![\tau']\!]$$

Interpretation of judgements:

$$[\![M]\!] : [\![\Gamma]\!] \to \mathbf{T}[\![\tau]\!]$$

$$\begin{aligned}
[\![x]\!]\rho &= \eta(\rho(x)) \\
[\![\lambda x \,.\, M]\!]\rho &= \eta(\lambda v \,.\, [\![M]\!]\rho\{x \mapsto v\}) \\
[\![M + N]\!]\rho &= [\![M]\!]\rho \gg= (\lambda m \,. \\
&\qquad [\![N]\!]\rho \gg= (\lambda n \,.\, \eta(m + n))) \\
[\![M \mathrm{~or~} N]\!]\rho &= \alpha([\![M]\!]\rho, [\![N]\!]\rho)
\end{aligned}$$

Powerset Semantics of λ_c^{or}

Interpretation of types:

$$[\![nat]\!] = \mathbb{N}, \quad [\![\tau \Rightarrow \tau']\!] = [\![\tau]\!] \Rightarrow \mathcal{P}[\![\tau']\!]$$

Interpretation of judgements:

$$[\![M]\!] : [\![\Gamma]\!] \to \mathcal{P}[\![\tau]\!]$$

$$\begin{aligned}
[\![x]\!]\rho &= \{\rho(x)\} \\
[\![\lambda x \,.\, M]\!]\rho &= \{\lambda v \,.\, [\![M]\!]\rho\{x \mapsto v\}\} \\
[\![M + N]\!]\rho &= \bigcup_{m \in [\![M]\!]\rho, n \in [\![N]\!]\rho} \{m + n\} \\
[\![M \,\mathrm{or}\, N]\!]\rho &= [\![M]\!]\rho \cup [\![N]\!]\rho
\end{aligned}$$

Part II

Relating Monadic Semantics

Relating Monadic Semantics

Semantics using $\mathcal{P}$

$$[\![M]\!] : [\![\Gamma]\!] \to \mathcal{P}[\![\tau]\!], \quad [\![M \text{ or } N]\!]\rho = [\![M]\!]\rho \cup [\![N]\!]\rho$$

$$[\![(3 \text{ or } 4) + (2 \text{ or } 3)]\!] = \{5, 6, 7\}$$

Relating Monadic Semantics

Semantics using $\mathcal{P}$

$$[\![M]\!] : [\![\Gamma]\!] \to \mathcal{P}[\![\tau]\!], \quad [\![M \text{ or } N]\!]\rho = [\![M]\!]\rho \cup [\![N]\!]\rho$$

$$[\![(3 \text{ or } 4) + (2 \text{ or } 3)]\!] = \{5, 6, 7\}$$

But wait — I learned to use the **list monad** to represent nondeterminism!

$$[\![M]\!] : [\![\Gamma]\!] \to [\![\tau]\!]^*, \quad [\![M \text{ or } N]\!]\rho = [\![M]\!]\rho \cdot [\![N]\!]\rho$$

$$[\![(3 \text{ or } 4) + (2 \text{ or } 3)]\!] = (5\ 6\ 6\ 7)$$

Relating Monadic Semantics

How about using the **continuation monad**?

$$\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \to (\llbracket \tau \rrbracket \Rightarrow \mathcal{P}R) \Rightarrow \mathcal{P}R,$$
$$\llbracket M \text{ or } N \rrbracket \rho = \lambda k\ .\ \llbracket M \rrbracket \rho k \cup \llbracket N \rrbracket \rho k$$

... or **2-continuation monad** [Wand&Vaillancourt'04]?

$$\llbracket M \rrbracket : \llbracket \Gamma \rrbracket \to (\llbracket \tau \rrbracket \Rightarrow X \Rightarrow X) \Rightarrow X \Rightarrow X,$$
$$\llbracket M \text{ or } N \rrbracket \rho = \lambda k\ .\ \llbracket M \rrbracket \rho k \circ \llbracket N \rrbracket \rho k$$

Relating Monadic Semantics

They interpret the same term differently:

$$\begin{aligned} [\![(3 \text{ or } 4) + (2 \text{ or } 3)]\!] &= \{5, 6, 7\} \\ &= (5\ 6\ 6\ 7) \\ &= \lambda k\ .\ k5 \cup k6 \cup k7 \\ &= \lambda k\ .\ k5 \circ k6 \circ k6 \circ k7 \end{aligned}$$

... but they appear to be related somehow.

Problem

How do we formally establish relationships between them?

Relating Monadic Semantics

There are many variations of this problem:

- Computational effects (**nondeterminism**, states, writer, I/O, ...)
- Monadic semantics ($\mathcal{P}, (-)^*, \mathcal{C}^{\mathcal{P}R}, \mathcal{C}^{X \Rightarrow X}, \cdots$)
- Their relationships

We want to solve a generalized problem!

Effect Simulation Problem

One language: an extension of the STLC with

$$\frac{\Gamma \vdash M_i : b_i \quad (1 \le i \le n)}{\Gamma \vdash op(M_1, \cdots, M_n) : b} \quad \frac{\Gamma \vdash M_i : \tau \quad (1 \le i \le n)}{\Gamma \vdash ef(M_1, \cdots, M_n) : \tau}$$

Two semantics: $[\![-]\!]_i$ using monad T_i $(i = 1, 2)$

$$[\![op]\!]_i : [\![b_1]\!]_i \times \cdots \times [\![b_n]\!]_i \to [\![b]\!]_i$$

$$[\![ef]\!]_{i,A} : T_i A \times \cdots \times T_i A \to T_i A$$

Relationship:

$$Vb \subseteq [\![b]\!]_1 \times [\![b]\!]_2, \quad Cb \subseteq T_1[\![b]\!]_1 \times T_2[\![b]\!]_2$$

Effect Simulation Problem

Effect Simulation Problem

For any

- $\Gamma = x_1 : b_1, \cdots, x_n : b_n$
- $\Gamma \vdash M : b$
- $\rho_1 \in [\![\Gamma]\!]_1, \rho_2 \in [\![\Gamma]\!]_2$ such that

$$(\rho_1(x_i), \rho_2(x_i)) \in Vb_i \quad (1 \le i \le n)$$

do we have

$$([\![M]\!]_1 \rho_1, [\![M]\!]_2 \rho_2) \in Cb?$$

Effect Simulation Problem

Theorem

The answer of the effect simulation problem is yes if

1. $([\![op]\!]_1, [\![op]\!]_2) : Vb_1 \dot{\times} \cdots \dot{\times} Vb_n \dot{\to} Vb$,
2. $([\![ef]\!]_{1,[\![b]\!]_1}, [\![ef]\!]_{2,[\![b]\!]_2}) : Cb \dot{\times} \cdots \dot{\times} Cb \dot{\to} Cb$ for all b,
3. $((\eta_1)_{[\![b]\!]_1}, (\eta_2)_{[\![b]\!]_2}) : Vb \dot{\to} Cb$ for all b.

Notation: for

- $R_i \subseteq A_i \times B_i$ and $S \subseteq C \times D$
- $f : A_1 \times \cdots \times A_n \to C$ and $g : B_1 \times \cdots \times B_n \to D$,

$(f, g) : R_1 \dot{\times} \cdots \dot{\times} R_n \dot{\to} S$ means

$$\forall \vec{x}, \vec{y} \,.\, (\forall 1 \leq i \leq n \,.\, (x_i, y_i) \in R_i) \implies (f\vec{x}, g\vec{y}) \in S$$

Proof

A candidate relational model

Construct $R\tau \subseteq [\![\tau]\!]_1 \times [\![\tau]\!]_2$ by induction:

$$Rb = Vb$$
$$R(\tau \Rightarrow \tau') = R\tau \mathrel{\dot{\Rightarrow}} \dot{T}(R\tau')$$

Here, for $R \subseteq A_1 \times A_2$ and $S \subseteq B_1 \times B_2$,

$$R \mathrel{\dot{\Rightarrow}} S = \{(f, g) \mid \forall (a, b) \in R \,.\, (fa, gb) \in S\}$$
$$\subseteq (A_1 \Rightarrow B_1) \times (A_2 \Rightarrow B_2)$$

Proof

A candidate relational model

Construct $R\tau \subseteq [\![\tau]\!]_1 \times [\![\tau]\!]_2$ by induction:

$$Rb = Vb$$
$$R(\tau \Rightarrow \tau') = R\tau \dot{\Rightarrow} \dot{\mathbf{T}}(R\tau')$$

Here, for $R \subseteq A_1 \times A_2$,

$$\dot{\mathbf{T}}R = \{(c, d) \mid \forall b \in B \,.\, \forall (f, g) \in R \dot{\Rightarrow} \mathbf{Cb} \,.\, (f^{\#1}(c), g^{\#2}(d)) \in \mathbf{Cb}\} \subseteq T_1 A_1 \times T_2 A_2$$

a semantic version of **⊤⊤-lifting** [Lindley&Stark'05]

Proof

For any $R \subseteq A_1 \times A_2$ and $S \subseteq B_1 \times B_2$,

Lemma

$(\eta_{1,A_1}, \eta_{2,A_2}) : R \dot{\to} \dot{T}R.$

Lemma

$(f, g) : R \dot{\to} \dot{T}S$ implies $(f^{\#1}, g^{\#2}) : \dot{T}R \dot{\to} \dot{T}S$

Lemma (proof uses assumption 2)

$$(\llbracket ef \rrbracket_{1,A_1}, \llbracket ef \rrbracket_{2,A_2}) : \dot{T}R \dot{\times} \cdots \dot{\times} \dot{T}R \dot{\to} \dot{T}R.$$

Proof

Lemma (R is indeed a relational model)

For any $x_1 : \tau_1, \cdots, x_n : \tau_n \vdash M : \tau$,

$$([\![M]\!]_1, [\![M]\!]_2) : R\tau_1 \dot{\times} \cdots \dot{\times} R\tau_n \dot{\to} \dot{T}R\tau.$$

Lemma (proof uses assumption 3)

For all $b \in B$, $\dot{T}(Vb) \subseteq Cb$.

Essense of Proof

... is to build the logical relation $\dot{T}$ for monads using Cb.

https://en.wikipedia.org/wiki/Aikido#/media/File:Shihonage.jpg

Part III

Effect Systems

Effect System

... extends type system to **estimate** side-effects caused by programs.

$$\Gamma \vdash M : \tau \ \& \ e$$

We read it as:

- The side-effect of M is at most **e**.
- M will not do anything outside the scope of **e**.

The latter is useful for effect-dependent program transformations.

Effect System [Luccassen&Gifford'88]

... extends system F with

$$\Gamma \vdash M : \tau \,\&\, e$$

where

$$e \in \mathcal{P}(rd(R) + wr(R) + al(R)) \quad \text{(as a join semilattice)}$$

and R is the set of **regions**.

$$M : \tau \,\&\, rd(\rho) \vee wr(\rho) \qquad N : \tau \,\&\, wr(\rho')$$

$\Longrightarrow$ M and N do not interfere with each other

Effect System [Luccassen&Gifford'88]

Allocation at region ρ:

$$\frac{M : \tau \ \& \ e}{\mathit{new}\ \rho\ \tau\ M : \mathit{ref}\ \rho\ \tau \ \& \ e \vee \mathit{al}(\rho)}$$

Read from region ρ:

$$\frac{M : \mathit{ref}\ \rho\ \tau \ \& \ e}{\mathit{get}\ M : \tau \ \& \ e \vee \mathit{rd}(\rho)}$$

Write to region ρ:

$$\frac{M : \mathit{ref}\ \rho\ \tau \ \& \ e \quad N : \tau' \ \& \ e' \quad \tau' \sqsubseteq \tau}{\mathit{set}\ M\ N : () \ \& \ e \vee e' \vee \mathit{wr}(\rho)}$$

Effect System

- Communication analysis in CML [Nielson&Nielson'93]
- Exception analysis of Java
- Effect-dependent program optimizations [Benton&Kennedy&Hofmann&Beringer'06]+, [Thamsborg&Birkedal'11]
- Session types and effects [Orchard&Yoshida'16]
- Cardinality analysis [Benton&Kennedy&Hofmann&Nigam'16]
- Cost analysis [Çiçek, Garg, Acar '17]
- ... and many more

A Simple Cardinality Analysis

Effects

$$E = (\mathbb{N}, \leq).$$

Types

$$\mathbf{Typ}^E \ni \tau ::= \mathit{nat} \mid \tau \overset{e}{\Rightarrow} \tau$$

Judgements

$$\Gamma \vdash M : \tau \;\&\; e$$

"*M* returns at most *e* choices"

A Simple Cardinality Analysis

Variable

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau \ \& \ 1}$$

Abstraction

$$\frac{\Gamma \vdash x : \tau \vdash M : \tau' \ \& \ e}{\Gamma \vdash \lambda x : \tau \, . \, M : \tau \overset{e}{\Rightarrow} \tau' \ \& \ 1}$$

Application

$$\frac{M : \tau \overset{e}{\Rightarrow} \tau' \ \& \ e' \qquad N : \tau \ \& \ e''}{MN : \tau' \ \& \ e'e''e}$$

A Simple Cardinality Analysis

Number

$$\frac{}{\Gamma \vdash n : \mathit{nat} \mathbin{\&} 1}$$

Addition

$$\frac{\Gamma \vdash M : \mathit{nat} \mathbin{\&} e \quad \Gamma \vdash N : \mathit{nat} \mathbin{\&} e'}{\Gamma \vdash M + N : \mathit{nat} \mathbin{\&} ee'}$$

Choice

$$\frac{\Gamma \vdash M : \tau \mathbin{\&} e \quad \Gamma \vdash N : \tau \mathbin{\&} e'}{\Gamma \vdash M \,\mathrm{or}\, N : \tau \mathbin{\&} e + e'}$$

Subeffecting

$$\frac{\Gamma \vdash M : \tau \mathbin{\&} e \quad e \leq e'}{\Gamma \vdash M : \tau \mathbin{\&} e'}$$

Effect Soundness

Problem: Effect soundness

Under the powerset semantics, for any $\emptyset \vdash M : \tau \ \& \ e$, do we have

$$|[\![M]\!]| \leq e?$$

Effect Soundness

There are many variations of this problem:

- Computational effects and operations on them
- Monadic semantics
- **Definition of effects**
- **Soundness statement**

We formulate a general effect soundness problem.

Generic Effect System

In many papers,

- Effects are **ordered**: to compare the extent / scope of effects.
- Effects are **composable**: to give the effect of the sequential execution.

$$\frac{M : nat \;\&\; e \qquad N : nat \;\&\; e'}{M + N : nat \;\&\; ee'}$$

The postulate on effects in this lecture

Effects form a preordered monoid.

$$\mathbb{E} = (E, \quad \lesssim, \quad 1 \in E, \quad (\cdot) : (E, \lesssim)^2 \to (E, \lesssim))$$

Generic Effect System

A preordered monoid of effects

$$(E, \leq, 1, \cdot).$$

Types

$$\mathbf{Typ}^E \ni \tau ::= nat \mid \tau \stackrel{e}{\Rightarrow} \tau$$

Effect erasure $|-| : \mathbf{Typ}^E \to \mathbf{Typ}$

$$|nat| = nat, \quad |\tau \stackrel{e}{\Rightarrow} \tau'| = |\tau| \Rightarrow |\tau'|$$

Judgements

$$\Gamma \vdash M : \tau \ \& \ e$$

Generic Effect System

Variable

$$\frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau \; \& \; 1}$$

Abstraction

$$\frac{\Gamma \vdash x : \tau \vdash M : \tau' \; \& \; e}{\Gamma \vdash \lambda x : \tau \,.\, M : \tau \stackrel{e}{\Rightarrow} \tau' \; \& \; 1}$$

Application

$$\frac{M : \tau \stackrel{e}{\Rightarrow} \tau' \; \& \; e' \qquad N : \tau \; \& \; e''}{MN : \tau' \; \& \; e' \cdot e'' \cdot e}$$

Generic Effect System

Base type operation

$$\frac{\Gamma \vdash M_i : b_i \,\&\, e_i \quad (1 \le i \le n)}{\Gamma \vdash op(M_1, \cdots, M_n) : b \,\&\, e_1 \cdot \ldots \cdot e_n}$$

Subeffecting

$$\frac{\Gamma \vdash M : \tau \,\&\, e \quad e \le e'}{\Gamma \vdash M : \tau \,\&\, e'}$$

Generic Effect System

Effectful Operation

$$\frac{\Gamma \vdash M_i : \tau \ \&\ e_i \quad (1 \le i \le n)}{\Gamma \vdash ef(M_1, \cdots, M_n) : \tau \ \&\ f(e_1, \cdots, e_n)}$$

where $f : E^n \to E$ is a **monotone function** such that

$$f(e_1, \cdots, e_n) \cdot e = f(e_1 \cdot e, \cdots, e_n \cdot e)$$

This reflects the axiom of algebraic operation:

$$\alpha(c_1, \cdots, c_n) \ggg k = \alpha(c_1 \ggg k, \cdots, c_n \ggg k)$$

Generic Effect Soundness

Definition

A **semantics of E** for T is an $E \times \mathbf{Typ}^E$-family of subsets $Ce\tau \subseteq T[\![|\tau|]\!]$, monotone on e.

Question: Effect Soundness

- $[\![-]\!]$: a monadic semantics using a monad T, ignoring effect annotations
- C: a semantics of E for T

For any $\emptyset \vdash M : \tau \;\&\; e$, do we have

$$[\![M]\!] \in Ce\tau?$$

Effect Soundness

Theorem

The effect soundness holds if

- $\eta_{[\![|\tau|]\!]} : [\![|\tau|]\!] \dot{\to} C1\tau$ for all $\tau \in \mathbf{Typ}^E$,
- $[\![ef]\!]_{[\![|\tau|]\!]} : Ce_1\tau \dot{\times} \cdots \dot{\times} Ce_n\tau \dot{\to} C(f(e_1, \cdots, e_n))\tau$ for all $\tau \in \mathbf{Typ}^E$ and $e_1, \cdots, e_n \in E$.

Notation (redefining):

- for $P_i \subseteq A_i$ and $S \subseteq B$
- for $f : A_1 \times \cdots \times A_n \to B$

$f : P_1 \dot{\times} \cdot \dot{\times} C_n \dot{\to} S$ means

$$\forall \vec{x} \,.\, (\forall 1 \leq i \leq n \,.\, x_i \in P_i) \implies f\vec{x} \in S$$

Proof

A candidate predicate model

Deifne $P\tau \subseteq [\![|\tau|]\!]$ (where $\tau \in \mathbf{Typ}^E$) inductively by

$$Pb = [\![b]\!], \quad P(\tau \overset{e}{\Rightarrow} \tau') = P\tau \dot{\Rightarrow} \dot{T}e(P\tau')$$

where for $e \in E$ and $X \subseteq A$, $\dot{T}eX$ is given by

$$\dot{T}eX = \{c \in TA \mid \forall d \in E \,.\, \forall \tau \in \mathbf{Typ}^E \,.\, \forall f \in X \dot{\Rightarrow} Cd\tau \,.\, f^{\#}c \in C(ed)\tau\}$$

Proof

Let $X \subseteq A$, $Y \subseteq B$ be subsets and $d, e_1, \cdots, e_n, e \in E$.

Lemma

$\eta_A : X \dot{\to} \dot{T}1X$.

Lemma

For any $f : X \dot{\to} \dot{T}eY$, we have $f^\# : \dot{T}dX \dot{\to} \dot{T}(de)Y$.

Lemma

$[\![ef]\!]_A : \dot{T}e_1X \dot{\times} \cdots \dot{\times} \dot{T}e_nX \dot{\to} \dot{T}(f(e_1, \cdots, e_n))X$.

Proof

Lemma

For any $x_1 : \tau_1, \cdots, x_n : \tau_n \vdash M : \tau \ \& \ e$,

$$[\![M]\!] : P\tau_1 \dot{\times} \cdots \dot{\times} P\tau_n \dot{\to} \dot{T}e(P\tau).$$

Lemma

For any $\tau \in Typ^E$, we have $\dot{T}e(P\tau) \subseteq \dot{T}e[\![|\tau|]\!] \subseteq Ce\tau$.

The goal is immediate from these two lemmata.