Semantics of Computational Effects and Effect Systems (Lecture 2)

Shin-ya Katsumata

National Institute of Informatics

Shonan school
17 May, 2017

Part I

Graded Monads

Properties of $\dot{T}$

The following properties are the key to construct a predicate model of the generic effect system:

Lemma

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

Lemma

$f : X \dot{\to} \dot{T}eY \implies f^{\#} : \dot{T}dX \dot{\to} \dot{T}(d \cdot e)Y.$

$\implies$ we introduce **graded monads**.

Graded Monad [Smirnov'08]

Let $(E, \lesssim, 1, \cdot)$ be a preordered monoid.

Definition

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

- T sends $e \in E$ and a set A to a set TeA.
- $T_{1,A} : A \to T1A$.
- $(-)^{e \# e'} : (A \Rightarrow Te'B) \to (TeA \Rightarrow T(e \cdot e')B)$.

Exercise

Guess the axioms of graded monads.

Graded Monad [Smirnov'08]

Proposition

Graded monad on **Set** $\simeq$ **lax monoidal functor**

$$T : \mathbb{E} \to ([\mathbf{Set}, \mathbf{Set}], \mathrm{Id}, \circ)$$

A **graded ring** R comes with an $\mathbb{N}$-indexed family R_i of abelian groups such that

$$\mathbf{R_i R_j \subseteq R_{i+j}}, \quad R = \bigoplus R_i$$

Such a family R_i forms a **lax monoidal functor**

$$R : (\mathbb{N}, 0, +) \to (\mathbf{Ab}, \mathbf{I}, \otimes)$$

Graded Writer Monad

Consider the preordered monoid of languages over Δ:

$$\mathbb{E} = (P(\Delta^*), \subseteq, \{\epsilon\}, \star) \quad (\star\text{: language concat.})$$

Graded Writer Monad:

$$\begin{array}{rcl} T & : & (P(\Delta^*), \subseteq) \to [\mathbf{Set}, \mathbf{Set}] \\ TeA & = & e \times A \end{array}$$

$$\begin{array}{rcl} T_{1,A} & : & A \to T\{\epsilon\}A \\ T_{1,A} & = & \lambda a \,.\, (\epsilon, a) \end{array}$$

$$\begin{array}{rcl} (-)^{e\#e'} & : & (A \Rightarrow Te'B) \to (TeA \Rightarrow T(e \star e')B) \\ f^{e\#e'}(a_1 \cdots a_n) & = & f(a_1) \cdots f(a_n) \end{array}$$

Cardinality-Graded Powerset Monad

Consider the ordered multiplicative monoid

$$(\mathbb{N}, \leq, 1, \times).$$

Cardinality-graded powerset monad:

$$\begin{aligned} \mathcal{P} &: (\mathbb{N}, \leq) \to [\mathbf{Set}, \mathbf{Set}] \\ \mathcal{P}eA &= \{X \subseteq A \mid card(X) \leq e\} \end{aligned}$$

$$\begin{aligned} \mathcal{P}_{1,A} &: A \to \mathcal{P}1A \\ \mathcal{P}_{1,A} &= \lambda a \,.\, \{a\} \end{aligned}$$

$$\begin{aligned} (-)^{e\#e'} &: (A \Rightarrow Te'B) \to (TeA \Rightarrow T(e \times e')B) \\ f^{e\#e'} &= \lambda X \,.\, \bigcup_{x \in X} f(x) \end{aligned}$$

Graded State and Continuation Monads

$$\begin{array}{ll} \mathbb{E} = (E, \lesssim, 1, \cdot) & \text{preordered monoid} \\ S : (E, \lesssim) \to \mathbf{Set} & \text{any functor} \end{array}$$

The following end is a graded monad for $\mathbb{E}$ on **Set**:

$$\begin{aligned} SeA &= \int_{d \in (E, \lesssim)} Sd \Rightarrow (A \times S(d \cdot e)) \\ CeA &= \int_{d \in (E, \lesssim)} (A \Rightarrow Sd) \Rightarrow S(e \cdot d) \end{aligned}$$

Graded State and Continuation Monads

$$\mathbb{E} = (E, \lesssim, 1, \cdot) \quad \text{preordered monoid}$$
$$S : (E, \lesssim) \to \mathbf{Set} \quad \text{any functor}$$

Theorem

For any parametrized monad [Atkey'09] $T : \mathbb{E}^{op} \times \mathbb{E} \times \mathbf{Set} \to \mathbf{Set}$, the following is a graded monad for $\mathbb{E}$.

$$TeA = \int_d T(d, d \cdot e, A),$$

Graded Monads for Join Semilattices

Let $(E, \leq)$ be a join semilattice.

Proposition

The following are isomorphic data:

- A graded monad for $(E, \leq, \bot, \vee)$
- A functor of type $(E, \leq) \to \mathbf{Monad}(\mathbf{Set})$

C.f. generalised monad [Fillâtre '99]

Alg. Ops. for Graded Monads

What is a good notion of algebraic operation for graded monads? Using the correspondece:

$$\{1,2\} \in \mathcal{P}\{1,2\} \iff x \cup y : (\mathcal{P}X)^2 \to \mathcal{P}X$$

we can derive:

$$\{1,2\} \in \mathcal{P}2\{1,2\} \iff x \cup y : (\mathcal{P}eX)^2 \to \mathcal{P}(2e)X,$$

but the latter is not very useful...

$$\frac{M : \tau \mathbin{\&} e \quad N : \tau \mathbin{\&} e}{M \text{ or } N : \tau \mathbin{\&} 2e}$$

Alg. Ops. for Graded Monads

The previous rule demands us to align the effect of each branch of or:

$$\dfrac{\dfrac{M : \tau \;\&\; 3}{M : \tau \;\&\; 6} \quad N : \tau \;\&\; 6}{M \;\mathrm{or}\; N : \tau \;\&\; 12}$$

We would rather like to have

$$\dfrac{M : \tau \;\&\; 3 \quad N : \tau \;\&\; 6}{M \;\mathrm{or}\; N : \tau \;\&\; 3 + 6}$$

Thus the effect of or should be computed by an **external** function, not by a single effect.

Alg. Ops. for Graded Monads

An **effect function** on E is a monotone function $\epsilon : (E, \leq)^n \to (E, \leq)$ such that

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

This reflects the equality

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

An effect function describes the effect of an algebraic operation.

Alg. Ops. for Graded Monads

Definition

An **(n,ϵ)-algebraic operation** of a graded monad T is

$$\alpha_{e_1,\cdots,e_n,A} : Te_1A \times \cdots \times Te_nA \to T(\epsilon(e_1,\cdots,e_n))A$$

such that for any $f : A \to TeB$,

$$\alpha(c \,{}^{e_1}\!\!\gg\!\!=^{e} f, \cdots, c \,{}^{e_n}\!\!\gg\!\!=^{e} f) = \alpha(c,\cdots,c) \,{}^{\epsilon(e_1,\cdots,e_n)}\!\!\gg\!\!=^{e} f$$

Example: $(\cup) : \mathcal{P}nA \times \mathcal{P}mA \to \mathcal{P}(n+m)A$

Part II

Constructing Graded Monads via Effect Observation

Effect Observation

It is often tedious to construct a graded monad by hand.

We present a construction via an **effect observation**:

$$O : T \longrightarrow (S, \sqsubseteq)$$

- $(S, \sqsubseteq)$ is a **preordered monad** over **Set** [K&Sato '13]
  — modeling an ordered algebra of effects
- T is a **Set**-monad
  — modeling side-effects of a language
- $O : T \to S$ is a monad morphism
  — modeling the **abstraction** of side-effects

Effect Observations

Given an effect observation:

$$O : T \longrightarrow (S, \sqsubseteq)$$

We construct a monoid $\mathbb{E}$:

$$\mathbb{E} = (S1, \sqsubseteq_1, 1, \star)$$

$$1 = 1 \xrightarrow{\eta_1} S1 \quad e \star e' = 1 \xrightarrow{e} S1 \xrightarrow{(e')^\#} S1$$

and a graded monad D:

$$DeA = \{c \in TA \mid O_1 \circ T!_A(c) \sqsubseteq_1 e\}$$

$$TA \xrightarrow{T!_A} T1 \xrightarrow{O_1} S1$$

Graded Writer Monad

From

$$\{-\} : Wr \longrightarrow (P \circ Wr, \subseteq)$$

We construct a monoid $\mathbb{E}$:

$$\mathbb{E} = (P(Wr1), \subseteq, 1, \star) \simeq (P(\Delta^*), \subseteq, \{\epsilon\}, \star)$$

and a graded monad D:

$$DeA = \{(w, c) \in \Delta^* \times A \mid \{(w, *)\} \in e\} \simeq e \times A$$

Graded Monad for Effect Analysis

Let Σ be a ranked alphabet.

$$\mathbb{E} = (P(|\Sigma| + 1), \cdots)$$

Meaning of effects:

- $\{f, g, *\}$: May perform f, g, or return a value
- $\{f, g, c\}$: May perform f, g, or c (but no return value)

$$DeA = \{c \in T_\Sigma A \mid ops(c) \subseteq e\}$$

$$ops(t) = \{o \mid o \text{ occurs in } t\} \cup \{* \mid t \text{ is not closed}\}$$

$$ops(f(x, c())) = \{f, c, *\} \quad ops(f(d(), c())) = \{f, c, d\}$$

Graded Monad for Effect Analysis

From

$$|-| : T_\Sigma \longrightarrow (P(|\Sigma| + -), \subseteq)$$

$$|x| = \{x\}, \quad |o(t_1, \cdots, t_n)| = \{o\} \cup |t_1| \cup \cdots \cup |t_n|$$

We construct a monoid $\mathbb{E}$:

$$\mathbb{E} = (P(|\Sigma| + 1), \subseteq, 1, \star)$$

whose multiplication is

$$\{f, g, *\} \star \{f, p, q, *\} = \{f, g, p, q, *\}$$

and a graded monad D:

$$DeA = \{c \in T_\Sigma A \mid |c[*/i]_{i \in A}| \subseteq e\}$$

Graded Monad for Effect Analysis

From

$$|-|: T_\Sigma \longrightarrow (P(|\Sigma| + -), \subseteq)$$

$$|x| = \{x\}, \quad |o(t_1, \cdots, t_n)| = \{o\} \cup |t_1| \cup \cdots \cup |t_n|$$

We construct a monoid $\mathbb{E}$:

$$\mathbb{E} = (P(|\Sigma| + 1), \subseteq, 1, \star)$$

whose multiplication is

$$\{f, g\} \star \{f, p, q, *\} = \{f, g\}$$

and a graded monad D:

$$DeA = \{c \in T_\Sigma A \mid |c[*/i]_{i \in A}| \subseteq e\}$$

Algebraic Operation

$$O : T \longrightarrow (S, \sqsubseteq)$$

The monad morphism O sends an n-ary algebraic operation α for T to the one $O\alpha$ for S.

Theorem

For any n-ary algebraic operation α for T, α restricts to $(n, O\alpha)$-algebraic operation for D:

$$\alpha_{e_1,\cdots,e_n,I} : De_1 I \times \cdots \times De_n I \to D(O\alpha(e_1, \cdots, e_n))I$$

Effect System **EFe**

A calculus **EFe** for $\mathbb{E} = (E, \lesssim, 1, \cdot)$ consists of:

- Type

$$\tau ::= b \mid \tau \Rightarrow \tau \mid \mathbf{T}e\tau \quad (b \in B, e \in E)$$

- **Explicit** subeffecting rule

$$\frac{\Gamma \vdash M : Te\tau \quad e \lesssim e'}{\Gamma \vdash T(e \lesssim e', M) : Te'\tau}$$

- Pure computation and sequential execution

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash [M] : T1\tau} \qquad \frac{\Gamma \vdash M : Te\tau \quad \Gamma, x : \tau \vdash N : Te'\sigma}{\Gamma \vdash \mathbf{let}^{e,e'}\ x\ \mathbf{be}\ M\ \mathbf{in}\ N : T(e \cdot e')\sigma}$$

Effect System **EFi**

A calculus **EFi** for $\mathbb{E} = (E, \lesssim, 1, \cdot)$ consists of:

- Type

$$\tau ::= b \mid \tau \Rightarrow \tau \mid \mathbf{T}e\tau \quad (b \in B, e \in E)$$

- **Implicit** subeffecting rule

$$\frac{\Gamma \vdash M : Te\tau \quad e \lesssim e'}{\Gamma \vdash M : Te'\tau}$$

- Pure computation and sequential execution

$$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash [M] : T1\tau} \quad \frac{\Gamma \vdash M : Te\tau \quad \Gamma, x : \tau \vdash N : Te'\sigma}{\Gamma \vdash \mathbf{let}\ x\ \mathbf{be}\ M\ \mathbf{in}\ N : T(e \cdot e')\sigma}$$

Refinement Semantics of **EFi**

- A graded monad over the CCC **Sub**(**Set**)
- A monad over the CCC **Set**

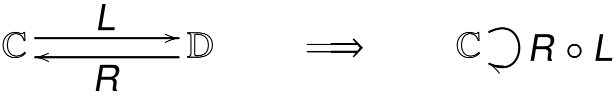
A semantics of **EFi** is given by

Part III

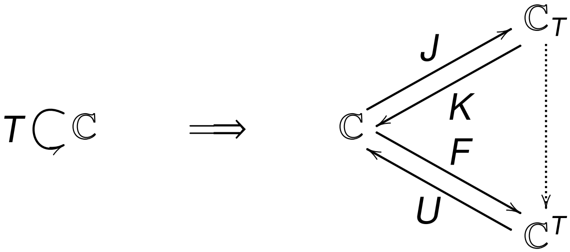
Resolutions of Graded Monads

Resolutions of Monads

An adjunction yields a monad:



A monad yields two adjunctions:



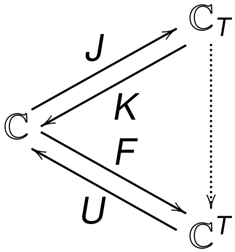
Resolutions of Monads

An adjunction **transports** a monad:

$$\mathbb{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathbb{D} \circlearrowright T \quad \Longrightarrow \quad \mathbb{C} \circlearrowright R \circ T \circ L$$

A monad yields two adjunctions:

$$T \circlearrowleft \mathbb{C} \quad \Longrightarrow$$



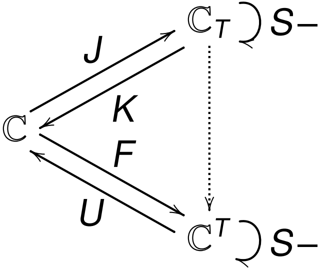
Resolutions of Graded Monads

An adjunction **transports** a graded monad:

$$\mathbb{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathbb{D} \circlearrowright T- \Longrightarrow \qquad \mathbb{C} \circlearrowright R \circ (T-) \circ L(=)$$

A graded monad yields two adjunctions with **twists**:

$$T- \circlearrowleft \mathbb{C} \qquad \Longrightarrow$$



Eilenberg-Moore Like resolution

... of a graded monad T uses the category $\mathbb{C}^T$ of **graded algebras** of T:

$$A : (E, \lesssim) \to \mathbb{C}, \quad a_{e,e'} : Te(Ae') \to A(ee').$$

$$\begin{array}{ccc} Ae & \xrightarrow{\eta} & T1(Ae) \\ & \searrow^{=} & \downarrow a \\ & & Ae \end{array} \qquad \begin{array}{ccc} Te(Te'(Ae'')) & \xrightarrow{Tea} & Te(A(e'e'')) \\ \mu \downarrow & & \downarrow a \\ T(ee')(Ae'') & \xrightarrow[a]{} & A(ee'e'') \end{array}$$

It comes with a **twist** functor $S : (E, \lesssim) \to [\mathbb{C}^T, \mathbb{C}^T]$:

$$S\mathbf{d}(A, a) = (\lambda e \,.\, A(e\mathbf{d}), \lambda e, e' \,.\, a_{e,e'\mathbf{d}})$$

Eilenberg-Moore Like resolution

... of a graded monad T uses the category $\mathbb{C}^T$ of **graded algebras** of T:

$$A : (E, \lesssim) \to \mathbb{C}, \quad a_{e,e'} : Te(Ae') \to A(ee').$$

$$\begin{array}{ccc} Ae & \xrightarrow{\eta} & T1(Ae) \\ & \diagdown\!\!\diagdown & \downarrow a \\ & & Ae \end{array} \qquad \begin{array}{ccc} Te(Te'(Ae'')) & \xrightarrow{Tea} & Te(A(e'e'')) \\ \mu\downarrow & & \downarrow a \\ T(ee')(Ae'') & \xrightarrow[a]{} & A(ee'e'') \end{array}$$

It comes with a **twist** functor $S : (E, \lesssim) \to [\mathbb{C}^T, \mathbb{C}^T]$:

$$S1 = \mathrm{Id}, \quad S(dd') = Sd \circ Sd'$$

Kleisli-Like Resolution

... of a graded monad T uses the following category $\mathbb{C}_T$:

Object (e, A) where $e \in \mathbb{E}, A \in \mathbb{C}$

Morphism The homset $\mathbb{C}_T((e, A), (e', B))$ is

$$\int^{d \in \mathbb{E}} \mathbb{E}(ed, e') \times \mathbb{C}(A, TdB)$$

It comes with a twist functor $S : (E, \lesssim) \to [\mathbb{C}_T, \mathbb{C}_T]$

$$S\mathbf{d}(e, A) = (e\mathbf{d}, A)$$

Kleisli-Like Resolution

... of a graded monad T uses the following category $\mathbb{C}_T$:

Object (e, A) where $e \in \mathbb{E}, A \in \mathbb{C}$

Morphism The homset $\mathbb{C}_T((e, A), (e', B))$ is

$$\int^{d \in \mathbb{E}} \mathbb{E}(ed, e') \times \mathbb{C}(A, TdB)$$

It comes with a twist functor $S : (E, \lesssim) \to [\mathbb{C}_T, \mathbb{C}_T]$

$$S1 = \mathrm{Id}, \quad S(dd') = Sd \circ Sd'$$

Category of Resolution of Graded Monads

Let T be an $\mathbb{E}$-graded monad on $\mathbb{C}$.

Object consists of

$$\mathbb{C} \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \mathbb{D} \circlearrowright S-$$

such that $S1 = \mathrm{Id}$ and $S(dd') = Sd \circ Sd'$

Morphism Map of adjunctions and twists.

Theorem (Fujii, K, Melliès ’16)

$\mathbb{C}_T$ and $\mathbb{C}^T$ are initial and final in the category of resolutions of T.

Action of a Monoidal Category

A monoid homomorphism

$$f : E \to (C \Rightarrow C, \mathrm{id}_C, \circ)$$

is nothing but a monoid action of E on C.
Its categorical analogue is a **strong** monoidal functor

$$F : \mathbb{E} \to ([\mathbb{C}, \mathbb{C}], \mathrm{Id}, \circ).$$

- Called **actegory** (McCrudden) / $\mathbb{E}$-category (Pareigis).
- Kelly and Janelidze studies when F has a right adjoint.

(Op)Lax Action of a Monoidal Category

For a **lax** monoidal $F : \mathbb{E} \to ([\mathbb{C}, \mathbb{C}], \mathrm{Id}, \circ)$,

- Durov employed it as a generalisation of **graded ring**.
- Smirnov studies it under the name **graded monad**.
- It is mentioned in [Atkey '08].
- A calculus similar to **EFe** / **EFi** appears in the APPSEM paper by Benton et al.
- Melliès studies it under the name **parametric monad** / **negative E-category**.

For an **oplax** monoidal $F : \mathbb{E} \to ([\mathbb{C}, \mathbb{C}], \mathrm{Id}, \circ)$,

- See recent work on **coeffects** [Petricek, Orchard, Mycroft '13], [Brunel, Gaboardi, Mazza, Zdancewic '14], [Ghica, Smith '14] and bounded LL.

Parameterisations of Monads