Flexible presentations of graded monads

Shin-ya Katsumata Dylan McDermott

Tarmo Uustalu Nicolas Wu

Motivation

1. Effects can be modelled using monads

   [Moggi '89]
2. which often come from presentations [Plotkin and Power '02]

3. which induce algebraic operations [Plotkin and Power '03]

Motivation

1. Effects can be modelled using monads [Moggi '89]
2. which often come from presentations [Plotkin and Power '02]
3. which induce algebraic operations [Plotkin and Power '03]

Example:

1. Nondeterminism can be modelled using List
2. which comes from the presentation of monoids

$$\mathsf{fail} : 0 \qquad \mathsf{or} : 2$$

$$\mathsf{or}(\mathsf{fail}, x) = x = \mathsf{or}(x, \mathsf{fail}) \qquad \mathsf{or}(\mathsf{or}(x, y), z) = \mathsf{or}(x, \mathsf{or}(y, z))$$

3. which induces algebraic operations

$$\mathrm{fail}_X = (\lambda\_.\,[\,]) \ : \ 1 \longrightarrow \mathrm{List}\,X$$

$$\mathrm{or}_X = (\lambda(\mathrm{xs}, \mathrm{ys}).\,\mathrm{xs} +\!\!+ \mathrm{ys}) \ : \ \mathrm{List}\,X \times \mathrm{List}\,X \longrightarrow \mathrm{List}\,X$$

Motivation

1. Effects with quantitative information can be modelled using graded monads

   [Katsumata '14]

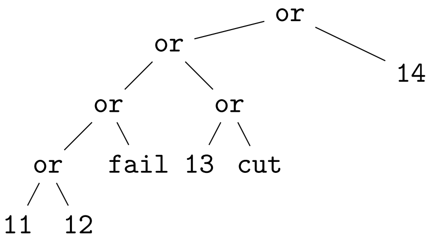
2. which often come from graded presentations?

   [Smirnov '08, Milius et al. '15, Dorsch et al. '19, Kura '20]

3. which induce algebraic operations?

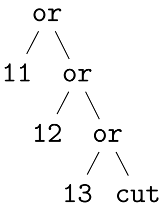
Running example: nondeterminism with backtracking and cut

```
or(or(or(or(return11, return12), fail),
         or(return13, cut)), return14)
```



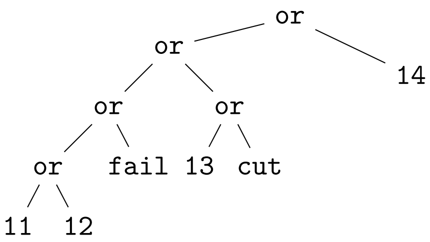
is equivalent to

```
or(return11,
   or(return12, or(return13, cut)))
```



Running example: nondeterminism with backtracking and cut

$$\mathtt{or}(\mathtt{or}(\mathtt{or}(\mathtt{or}(\mathtt{return}\,11, \mathtt{return}\,12), \mathtt{fail}), \mathtt{or}(\mathtt{return}\,13, \mathtt{cut})), \mathtt{return}\,14)$$



These computations can be modelled using a monad Cut

$$\mathrm{Cut}\,X = \mathrm{List}\,X \times \{\mathsf{cut}, \mathsf{nocut}\}$$

which has a presentation involving $\mathtt{or} : 2$, $\mathtt{fail} : 0$, $\mathtt{cut} : 0$ [Piróg and Staton '17]

Running example: nondeterminism with backtracking and cut

$$\texttt{or}(t, u) \equiv t \qquad \text{if } t \texttt{ cuts}$$

Running example: nondeterminism with backtracking and cut

Assign grades $e \in \{\bot, 1, \top\}$ to computations:

$\top$ don't know anything

$\mathrel{\rotatebox{90}{$\leq$}}$

1 definitely cuts or returns something

$\mathrel{\rotatebox{90}{$\leq$}}$

$\bot$ definitely cuts

$$\frac{}{\texttt{return}\,x \text{ has grade } 1}$$

$$\frac{t \text{ has grade } e \qquad e \leq e'}{t \text{ has grade } e'}$$

$$\frac{t_1 \text{ has grade } e_1 \qquad t_2 \text{ has grade } e_2}{\texttt{or}(t_1, t_2) \text{ has grade } (e_1 \sqcap e_2)}$$

$$\frac{}{\texttt{fail} \text{ has grade } \top}$$

$$\frac{}{\text{cut has grade } \bot}$$

Then:

$$\texttt{or}(t, u) \equiv t \qquad \text{if } t \text{ has grade } \bot$$

Running example: nondeterminism with backtracking and cut

Assign grades $e \in \{\bot, 1, \top\}$ to computations:

$\top$ don't know anything

$\geq$

1 definitely cuts or `returns` something

$\geq$

$\bot$ definitely cuts

Graded monad Cut:

$$\mathrm{Cut} X e = \{(\mathrm{xs}, c) \in \mathrm{List} X \times \{\mathsf{cut}, \mathsf{nocut}\} \mid (e = \bot \Rightarrow c = \mathsf{cut}) \wedge (e = 1 \Rightarrow c = \mathsf{cut} \vee \mathrm{xs} \neq [])\}$$

Kleisli extension:

$$\frac{f : X \to \mathrm{Cut} Y e}{f_d^\dagger : \mathrm{Cut} X d \to \mathrm{Cut} Y (d \cdot e)} \quad \text{where} \quad \begin{array}{rcl} \top \cdot e & = & \top \\ 1 \cdot e & = & e \\ \bot \cdot e & = & \bot \end{array}$$

Rigidly graded presentations [Smirnov '08, Milius et al. '15, Dorsch et al. '19, Kura '20]

Each operation op has an arity $n \in \mathbb{N}$ and grade d

$$\frac{t_1 \text{ has grade } e \qquad \cdots \qquad t_n \text{ has grade } e}{\mathrm{op}(t_1, \ldots, t_n) \text{ has grade } d \cdot e}$$

Rigidly graded presentations [Smirnov '08, Milius et al. '15, Dorsch et al. '19, Kura '20]

Each operation op has an arity $n \in \mathbb{N}$ and grade d

$$\frac{t_1 \text{ has grade } e \qquad \cdots \qquad t_n \text{ has grade } e}{\mathrm{op}(t_1, \ldots, t_n) \text{ has grade } d \cdot e}$$

These work well mathematically, but:

$$\frac{t_1 \text{ has grade } e_1 \qquad t_2 \text{ has grade } e_2}{\texttt{or}(t_1, t_2) \text{ has grade } (e_1 \sqcap e_2)} \quad ???$$

For or, we must have $d \geq 1$, but then $\texttt{or}(\texttt{cut}, \texttt{return}\, 14)$ will not have grade $\perp$

Flexibly graded presentations

$$\frac{t_1 \text{ has grade } d'_1 \cdot e \qquad \cdots \qquad t_n \text{ has grade } d'_n \cdot e}{\mathrm{op}(t_1, \ldots, t_n) \text{ has grade } d \cdot e}$$

$$\frac{t_1 \text{ has grade } e_1 \qquad t_2 \text{ has grade } e_2}{\mathtt{or}(t_1, t_2) \text{ has grade } (e_1 \sqcap e_2)}$$

Grading

Have an ordered monoid $(\mathbb{E}, 1, \cdot, \leq)$ of grades $d, e \in \mathbb{E}$:

- a monoid $(\mathbb{E}, 1, \cdot)$
- with a partial order $\leq$ on $\mathbb{E}$
- such that $(\cdot) : \mathbb{E} \times \mathbb{E} \to \mathbb{E}$ is monotone

Examples:

- Nondeterminism with cut: $(\mathbb{E}, \leq) = \{\bot \leq 1 \leq \top\}$

$$\begin{aligned} \top \cdot e &= \top \\ 1 \cdot e &= e \\ \bot \cdot e &= \bot \end{aligned}$$

- Gifford-style effect systems: $(\mathcal{P}\{\mathtt{get}, \mathtt{put}, \mathtt{raise}, \ldots\}, \emptyset, \cup, \subseteq)$

Flexibly graded presentations

Syntax:

- a flexibly graded signature is a collection of operations
- given a signature Σ, generate terms

$$x_1 : d'_1, \ldots, x_n : d'_n \vdash t : d$$

- a flexibly graded presentation is a signature Σ, with a collection E of equations
- given a presentation (Σ, E), have an equational logic

$$\Gamma \vdash t \equiv u : d$$

Semantics $\rightsquigarrow$ graded monads

Terms and substitution

Terms in context:

$$x_1 : d'_1, \ldots, x_n : d'_n \vdash t : d$$

Variables:

$$\frac{}{x_1 : d'_1, \ldots, x_n : d'_n \vdash x_i : d'_i}$$

Substitution:

$$\frac{x_1 : d'_1, \ldots, x_n : d'_n \vdash t : d \qquad \Gamma \vdash u_1 : d'_1 \cdot e \qquad \cdots \qquad \Gamma \vdash u_n : d'_n \cdot e}{\Gamma \vdash t\{e; x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\} : d \cdot e}$$

A special case:

$$\frac{x_1 : 1, \ldots, x_n : 1 \vdash t : d \qquad \Gamma \vdash u_1 : e \qquad \cdots \qquad \Gamma \vdash u_n : e}{\Gamma \vdash t\{e; x_1 \mapsto u_1, \ldots, x_n \mapsto u_n\} : d \cdot e}$$

$$\frac{f : [n] \to \mathrm{Cut}Y e}{f^\dagger_d : \mathrm{Cut}\,[n]\, d \to \mathrm{Cut}Y(d \cdot e)}$$

Flexibly graded signatures

Definition

A *flexibly graded signature* consists of a set

$$\Sigma(d_1', \ldots, d_n'; d)$$

for each $d_1', \ldots, d_n', d \in \mathbb{E}$.

Example

$$\mathsf{or}_{d_1,d_2} \in \Sigma(d_1, d_2; (d_1 \sqcap d_2)) \qquad \text{(for each } d_1, d_2 \in \mathbb{E}\text{)}$$

$$\mathsf{fail} \in \Sigma(\ ; \top)$$

$$\mathsf{cut} \in \Sigma(\ ; \bot)$$

Terms

Given a signature Σ, generate terms by

$$\frac{(x : d) \in \Gamma}{\Gamma \vdash x : d} \qquad \frac{d \leq d' \quad \Gamma \vdash t : d}{\Gamma \vdash (d{\leq}d')^* \, t : d'}$$

$$\frac{\mathrm{op} \in \Sigma(d'_1, \ldots, d'_n; d) \qquad \Gamma \vdash t_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash t_n : d'_n \cdot e}{\Gamma \vdash \mathrm{op}(e; t_1, \ldots, t_n) : d \cdot e}$$

Substitution:

$$\begin{aligned}(\mathrm{op}(e; t_1, \ldots, t_n))\{e'; x_1 \mapsto u_1, \ldots\} \\ = \ \mathrm{op}(e{\cdot}e'; t_1\{e'; x_1 \mapsto u_1, \ldots\}, \ldots, t_n\{e'; x_1 \mapsto u_1, \ldots\})\end{aligned}$$

Terms

Given a signature Σ, generate terms by

$$\frac{(x : d) \in \Gamma}{\Gamma \vdash x : d} \qquad \frac{d \leq d' \quad \Gamma \vdash t : d}{\Gamma \vdash (d{\leq}d')^* \, t : d'}$$

$$\frac{\mathrm{op} \in \Sigma(d'_1, \ldots, d'_n; d) \qquad \Gamma \vdash t_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash t_n : d'_n \cdot e}{\Gamma \vdash \mathrm{op}(e; t_1, \ldots, t_n) : d \cdot e}$$

Example

$$\frac{\Gamma \vdash t_1 : d'_1 \cdot e \qquad \Gamma \vdash t_2 : d'_2 \cdot e}{\Gamma \vdash \ \mathsf{or}_{d'_1,d'_2}(e; t_1, t_2) \ : (d'_1 \sqcap d'_2) \cdot e \ (= (d'_1 \cdot e) \sqcap (d'_2 \cdot e))} \qquad (\mathsf{or}_{d'_1,d'_2} \in \Sigma(d'_1, d'_2; (d'_1 \sqcap d'_2)))$$

$$\frac{}{\Gamma \vdash \mathsf{fail}(e; \ ) : \top \cdot e \ (= \top)} \qquad (\mathsf{fail} \in \Sigma(\ ; \top))$$

$$\frac{}{\Gamma \vdash \mathsf{cut}(e; \ ) : \bot \cdot e \ (= \bot)} \qquad (\mathsf{cut} \in \Sigma(\ ; \bot))$$

Flexibly graded presentations

Definition

A flexibly graded presentation consists of

- a signature Σ
- for each $d'_1, \ldots, d'_n, d \in \mathbb{E}$, a set $E(d'_1, \ldots, d'_n;\ d)$ of equations

$$x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d$$

Example

$$x : e_1 \cdot d, y : e_2 \cdot d \vdash \mathsf{or}_{e_1,e_2}(d; x, y) \equiv \mathsf{or}_{e_1 \cdot d, e_2 \cdot d}(1; x, y) : (e_1 \sqcap e_2) \cdot d$$

$$x : e_1, y : e_2 \vdash (e_1 \sqcap e_2 \leq e'_1 \sqcap e'_2)^*(\mathsf{or}_{e_1,e_2}(1; x, y)) \equiv \mathsf{or}_{e_1,e_2}(1; (e_1 \leq e'_1)^* x, (e_2 \leq e'_2)^* y) : e'_1 \sqcap e'_2$$

$$x : e \vdash \mathsf{or}_{\top,e}(1; \mathsf{fail}(1;\ ), x) \equiv x : e \qquad x : e \vdash x \equiv \mathsf{or}_{e,\top}(1; x, \mathsf{fail}(1;\ )) : e$$

$$x : e_1, y : e_2, z : e_3 \vdash \mathsf{or}_{e_1 \sqcap e_2, e_3}(1; \mathsf{or}_{e_1,e_2}(1; x, y), z) \equiv \mathsf{or}_{e_1, e_2 \sqcap e_3}(1; x, \mathsf{or}_{e_2,e_3}(1; y, z)) : e$$

$$x : \bot, y : e \vdash \mathsf{or}_{\bot,e}(1; x, y) \equiv x : \bot$$

Example: stacks of booleans

A grading of a presentation from [Goncharov '13]:

- Grades: $(\mathbb{N}, 0, +, \leq)$ (has grade $e \in \mathbb{N}$ = pushes at most e values)
- Operations:

$$\mathsf{push}_v \in \Sigma(0; 1) \qquad \frac{\Gamma \vdash t : e}{\Gamma \vdash \mathsf{push}_v(e; t) : 1 + e} \qquad (v \in \{\mathsf{true}, \mathsf{false}\})$$

$$\mathsf{pop} \in \Sigma(0, 1, 1; 0) \qquad \frac{\Gamma \vdash t_{\mathsf{empty}} : e \qquad \Gamma \vdash u_{\mathsf{true}} : 1 + e \qquad \Gamma \vdash u_{\mathsf{false}} : 1 + e}{\Gamma \vdash \mathsf{pop}(e; t_{\mathsf{empty}}, u_{\mathsf{true}}, u_{\mathsf{false}}) : e}$$

- Equations:

$$\mathsf{push}_{\mathsf{true}}(0; \mathsf{pop}(0; x, y_{\mathsf{true}}, y_{\mathsf{false}})) \equiv y_{\mathsf{true}}$$

$$\mathsf{push}_{\mathsf{false}}(0; \mathsf{pop}(0; x, y_{\mathsf{true}}, y_{\mathsf{false}})) \equiv y_{\mathsf{false}}$$

$$\mathsf{pop}(0; x, \mathsf{push}_{\mathsf{true}}(0; x), \mathsf{push}_{\mathsf{false}}(0; x)) \equiv x$$

$$\mathsf{pop}(0; \mathsf{pop}(0; x, y_{\mathsf{true}}, y_{\mathsf{false}}), z_{\mathsf{true}}, z_{\mathsf{false}}) \equiv \mathsf{pop}(0; x, z_{\mathsf{true}}, z_{\mathsf{false}})$$

Flexibly graded equational logic

Generate

$$\Gamma \vdash t \equiv u : d$$

by reflexivity, transitivity, symmetry, congruence, naturality of operations, functoriality of $(-)^*$, and

$$\frac{(t, u) \in E(d'_1, \ldots, d'_n; d) \qquad \Gamma \vdash s_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash s_n : d'_n \cdot e}{\Gamma \vdash t\{e; x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\} \equiv u\{e; x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\} : d \cdot e}$$

Example: using $\mathsf{push}_{\mathsf{true}}(0; \mathsf{pop}(0; x, y_{\mathsf{true}}, y_{\mathsf{false}})) \equiv y_{\mathsf{true}}$ we have

$$\frac{\Gamma \vdash t : e \quad \Gamma \vdash u_{\mathsf{true}} : 1 + e \quad \Gamma \vdash u_{\mathsf{false}} : 1 + e}{\Gamma \vdash \mathsf{push}_{\mathsf{true}}(e; \mathsf{pop}(e; t, u_{\mathsf{true}}, u_{\mathsf{false}})) \equiv u_{\mathsf{true}} : 1 + e}$$

Graded sets

Definition

A graded set X is a functor $X : (\mathbb{E}, \leq) \longrightarrow \mathbf{Set}$:

- a set Xe for each $e \in \mathbb{E}$ (elements of X of grade e)
- a function $(e{\leq}e')^* : Xe \longrightarrow Xe'$ for each $e \leq e' \in \mathbb{E}$

such that $X(e \leq e) = \mathrm{id}$ and $X(e'{\leq}e'') \circ X(e{\leq}e') = X(e{\leq}e'')$.

Example: for each presentation (Σ, E) and context Γ

$$\mathrm{Tm}_{(\Sigma,E)}\Gamma e \;=\; \{[t]_{\equiv} \mid \Gamma \vdash t : e\}$$

Graded monads

Definition (Smirnov '08, Melliès '12, Katsumata '14)

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

- a graded set TX for each (ungraded) set X
- unit functions $\eta_X : X \to TX1$
- Kleisli extension $\dfrac{f : X \to TYe}{f^\dagger_d : TXd \to TY(d \cdot e)}$ natural in d, e

satisfying some laws

Example

Cut is a graded monad:

$$\begin{aligned}\mathrm{Cut}\,X\,e = \{(\mathrm{xs}, c) \in \mathrm{List}X \times \{\mathsf{cut}, \mathsf{nocut}\} \\ \mid (e = \bot \Rightarrow c = \mathsf{cut}) \\ \wedge (e = 1 \Rightarrow c = \mathsf{cut} \vee \mathrm{xs} \neq [])\}\end{aligned}$$

$$\begin{aligned}\eta_X x &= ([x], \mathsf{nocut}) \\ f^\dagger_d([x_1, \ldots, x_n], c) &= f\,x_1 \oplus \cdots \oplus f\,x_n \oplus ([], c) \\ (\mathrm{ys}, \mathsf{cut}) \oplus (\mathrm{ys}', c) &= (\mathrm{ys}, \mathsf{cut}) \\ (\mathrm{ys}, \mathsf{nocut}) \oplus (\mathrm{ys}', c) &= (\mathrm{ys} \mathbin{+\!\!+} \mathrm{ys}', c)\end{aligned}$$

Algebraic operations

Definition

A $(d'_1, \ldots, d'_n; d)$-ary algebraic operation for a graded monad T is a family of functions

$$\alpha_{X,e} : \textstyle\prod_i TX(d'_i \cdot e) \to TX(d \cdot e)$$

natural in e and satisfying

$$f^{\dagger}_{d \cdot e}(\alpha_{X,e}(t_1, \ldots, t_n)) = \alpha_{Y,e \cdot e'}(f^{\dagger}_{d'_1 \cdot e} t_1, \ldots, f^{\dagger}_{d'_n \cdot e} t_n) \qquad (f : X \to TYe')$$

Example

For the graded monad Cut, we have

$$\llbracket \mathsf{or}_{d'_1, d'_2} \rrbracket_{X,e} = (\oplus) : \mathrm{Cut}X(d'_1 \cdot e) \times \mathrm{Cut}X(d'_2 \cdot e) \to \mathrm{Cut}X((d'_1 \sqcap d'_2) \cdot e)$$

$$\llbracket \mathsf{fail} \rrbracket_{X,e} = (\lambda \_. \, ([], \mathsf{nocut})) : 1 \to \mathrm{Cut}X(\top \cdot e)$$

$$\llbracket \mathsf{cut} \rrbracket_{X,e} = (\lambda \_. \, ([], \mathsf{cut})) : 1 \to \mathrm{Cut}X(\bot \cdot e)$$

Presenting graded monads

Given a flexibly graded presentation (Σ, E), we want

- a graded monad $T_{(\Sigma,E)}$
- with a $(d'_1, \ldots, d'_n; d)$-ary algebraic operation

$$[\![\mathrm{op}]\!]_{X,e} : \prod_i T_{(\Sigma,E)} X(d'_i \cdot e) \longrightarrow T_{(\Sigma,E)} X(d \cdot e)$$

  for each $\mathrm{op} \in \Sigma(d'_1, \ldots, d'_n; d)$ (satisfying equations)
- that is in some sense canonical

Algebras

If (Σ, E) is a flexibly graded presentation, a (Σ, E)-algebra $(A, [\![-]\!])$ is

- a graded set A
- with a natural family of functions

$$[\![\mathrm{op}]\!]_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $\mathrm{op} \in \Sigma(d'_1, \dots, d'_n; d)$
- such that

$$[\![t]\!]_e = [\![u]\!]_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $e \in \mathbb{E}$ and axiom $x_1 : d'_1, \dots, x_n : d'_n \vdash t \equiv u : d$

Example

- $T_{(\Sigma,E)}X$, with algebraic operations $[\![\mathrm{op}]\!]_X$
- $\mathrm{Tm}_{(\Sigma,E)}\Gamma$, with $[\![\mathrm{op}]\!]_e([t_1]_\equiv, \dots, [t_n]_\equiv) = [\mathrm{op}(e; t_1, \dots, t_n)]_\equiv$

The equational logic is sound and complete:

$$\Gamma \vdash t \equiv u : d \quad \Leftrightarrow \quad \text{for all } (\Sigma, E)\text{-algebras } (A, [\![-]\!]),\ \ [\![t]\!] = [\![u]\!]$$

Algebras

If (Σ, E) is a flexibly graded presentation, a (Σ, E)-algebra $(A, [\![-]\!])$ is

- a graded set A
- with a natural family of functions

$$[\![\mathrm{op}]\!]_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $\mathrm{op} \in \Sigma(d'_1, \ldots, d'_n; d)$
- such that

$$[\![t]\!]_e = [\![u]\!]_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $e \in \mathbb{E}$ and axiom $x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d$

A morphism $f : (A, [\![-]\!]) \to (A', [\![-]\!]')$ is a natural family of functions

$$f_d : Ad \to A'd$$

preserving $[\![\mathrm{op}]\!]$

Algebras

If (Σ, E) is a flexibly graded presentation, a (Σ, E)-algebra $(A, [\![-]\!])$ is

- a graded set A
- with a natural family of functions

$$[\![\mathrm{op}]\!]_e : \textstyle\prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $\mathrm{op} \in \Sigma(d'_1, \ldots, d'_n; d)$
- such that

$$[\![t]\!]_e = [\![u]\!]_e : \textstyle\prod_i A(d'_i \cdot e) \to A(d \cdot e)$$

  for each $e \in \mathbb{E}$ and axiom $x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d$

A morphism $f : (A, [\![-]\!]) \xrightarrow{e} (A', [\![-]\!]')$ of grade e is a natural family of functions

$$f_d : Ad \to A'(d \cdot e)$$

preserving $[\![\mathrm{op}]\!]$

Locally graded categories [Wood '76]

Definition

A *locally graded category* $\mathcal{C}$ consists of

- a collection $|\mathcal{C}|$ of objects
- graded sets $\mathcal{C}(X, Y)$ of morphisms $\quad (f : X -e\to Y$ means $f \in \mathcal{C}(X, Y)e)$
- identities $\mathrm{id}_X : X -1\to X$
- composition

$$\frac{f : X -e\to Y \qquad g : Y -e'\to Z}{g \circ f : X -e \cdot e'\to Z}$$

  natural in e, e'

such that

$$\mathrm{id}_Y \circ f = f = f \circ \mathrm{id}_X \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

(These are categories enriched over $[\mathbb{E}, \mathbf{Set}]$ with Day convolution)

Locally graded categories

Every graded monad T has a locally graded Kleisli category $\mathbf{Kl}(\mathrm{T})$:

- Objects are sets X
- Morphisms $f : X \overset{e}{\rightarrow} Y$ are functions $f : X \to TYe$

The locally graded category $\mathbf{GSet}$:

- Objects are graded sets
- Morphisms $f : X \overset{e}{\rightarrow} Y$ are families of functions $f_d : Xd \to Y(d \cdot e)$, natural in d
- Identities are the identity functions
- Composition $g \circ f$ is

$$(g \circ f)_d : Xd \xrightarrow{f_d} Y(d \cdot e) \xrightarrow{g_{d \cdot e}} Z(d \cdot e \cdot e')$$

$\mathbf{Alg}(\Sigma, E)$:

- Objects are (Σ, E)-algebras
- Morphisms are as in $\mathbf{GSet}$, but preserving $[\![-]\!]$
- Identities and composition: as in $\mathbf{GSet}$

Functors

Definition

A *functor* $F : \mathcal{C} \longrightarrow \mathcal{D}$ between locally graded categories is an object mapping $F : |\mathcal{C}| \longrightarrow |\mathcal{D}|$ with a mapping of morphisms

$$\frac{f : X \overset{e}{\rightarrow} Y}{Ff : FX \overset{e}{\rightarrow} FY}$$

natural in e, and preserving identities and composition.

There is a forgetful functor

$$\begin{aligned} U_{(\Sigma,E)} : \mathbf{Alg}(\Sigma, E) &\longrightarrow \mathbf{GSet} \\ (A, [\![-]\!]) &\mapsto A \\ f &\mapsto f \end{aligned}$$

Algebra

An (Eilenberg-Moore) algebra for a graded monad T is

- a graded set A
- with an extension operator

$$\frac{f : X \to Ae}{f_d^{\ddagger} : TXd \to A(d \cdot e)}$$

- satisfying some laws

These form a locally graded category, with a forgetful functor:

$$U_{\mathsf{T}} : \mathbf{EM}(\mathsf{T}) \to \mathbf{GSet}$$

Presenting graded monads

Theorem

For every flexibly graded presentation (Σ, E), *there is*

- *a graded monad* $\mathsf{T}_{(\Sigma,E)}$
- *and functor* $R_{(\Sigma,E)} : \mathbf{Alg}(\Sigma, E) \longrightarrow \mathbf{EM}(\mathsf{T}_{(\Sigma,E)})$ *over* $\mathbf{GSet}$

such that

- $(T_{(\Sigma,E)}X, (-)^{\dagger}) = R_{(\Sigma,E)}(T_{(\Sigma,E)}X, [\![-]\!]_X)$ *for some* $[\![-]\!]_X$
- *for every graded monad* T' *and functor* $R' : \mathbf{Alg}(\Sigma, E) \longrightarrow \mathbf{EM}(\mathsf{T}')$ *over* $\mathbf{GSet}$, *there is a unique* $F : \mathbf{EM}(\mathsf{T}_{(\Sigma,E)}) \longrightarrow \mathbf{EM}(\mathsf{T}')$ *over* $\mathbf{GSet}$ *such that*

$$\begin{array}{ccc} \mathbf{Alg}(\Sigma, E) & \xrightarrow{R_{(\Sigma,E)}} & \mathbf{EM}(\mathsf{T}_{(\Sigma,E)}) \\ & \searrow^{R'} & \downarrow F \\ & & \mathbf{EM}(\mathsf{T}') \end{array}$$

Presenting graded monads

For the presentation of nondeterminism with Cut

$$\mathsf{T}_{(\Sigma,E)} \cong \mathsf{Cut}$$

with algebraic operations $[\![\mathsf{cut}_{d'_1,d'_2}]\!]$, $[\![\mathsf{fail}]\!]$, $[\![\mathsf{cut}]\!]$

For the presentation of stacks of booleans:

$$T_{(\Sigma,E)}Xe \cong \mathrm{Stk}Xe \cong \{t : \mathrm{List}\,2 \to \mathrm{List}\,2 \times X \mid (\forall \mathrm{vs}.\, |\mathrm{fst}(t\,\mathrm{vs})| \leq |\mathrm{vs}| + e) \wedge \cdots\}$$

$$[\![\mathsf{push}_v]\!]_{X,e} : \mathrm{Stk}Xe \to \mathrm{Stk}X(1+e)$$
$$[\![\mathsf{push}_v]\!]_{X,e}\, t\, \mathrm{vs} = t(v :: \mathrm{vs})$$

$$[\![\mathsf{pop}]\!]_{X,e} : \mathrm{Stk}Xe \times \mathrm{Stk}X(1+e) \times \mathrm{Stk}X(1+e) \to \mathrm{Stk}Xe$$

$$[\![\mathsf{pop}]\!]_{X,e}(t_{\mathsf{empty}}, u_{\mathsf{true}}, u_{\mathsf{false}})\ \mathrm{vs} = \begin{cases} t_{\mathsf{empty}}\ [] & \text{if vs} = [] \\ u_{\mathrm{head\,vs}}\ (\mathrm{tail\,vs}) & \text{otherwise} \end{cases}$$

Constructing $T_{(\Sigma,E)}$

flexibly graded presentations

$\downarrow$ $(\Sigma,E) \mapsto \mathrm{Tm}_{(\Sigma,E)}$

flexibly graded clones $=$ sets of terms, with variables and substitution

$\downarrow$ left Kan extension along $\mathbf{FCtx} \to \mathbf{GSet}$

flexibly graded monads $=$ monad on $\mathbf{GSet}$

$\downarrow$ compose with $\mathbf{Free}(\mathbf{Set}) \to \mathbf{GSet}$

graded monads

Constructing $T_{(\Sigma,E)}$

algebraic theories and relative monads are closely connected (jww Nathanael Arkor)

flexibly graded presentations

$(\Sigma,E) \mapsto \mathrm{Tm}_{(\Sigma,E)}$ $\downarrow \simeq \uparrow$

flexibly graded clones = $(\mathbf{FCtx} \to \mathbf{GSet})$-relative monad

left Kan extension along $\mathbf{FCtx} \to \mathbf{GSet}$ $\downarrow \simeq \uparrow$ compose with $\mathbf{FCtx} \to \mathbf{GSet}$

flexibly graded monads preserving conical sifted colimits = monad on $\mathbf{GSet}$ preserving conical sifted colimits

$\uparrow \dashv \downarrow$ compose with $\mathbf{Free}(\mathbf{Set}) \to \mathbf{GSet}$

graded monads preserving conical sifted colimits = $(\mathbf{Free}(\mathbf{Set}) \to \mathbf{GSet})$-relative monad preserving conical sifted colimits

Given a flexibly graded presentation (Σ, E), there is

- a graded monad $T_{(\Sigma,E)}$
- with a $(d'_1, \dots, d'_n; d)$-ary algebraic operation

$$[\![\mathrm{op}]\!]_{X,e} : \prod_i T_{(\Sigma,E)}X(d'_i \cdot e) \to T_{(\Sigma,E)}X(d \cdot e)$$

  for each $\mathrm{op} \in \Sigma(d'_1, \dots, d'_n; d)$ (satisfying equations)
- that is in some sense canonical

Every graded monad that preserves conical sifted colimits has a flexibly graded presentation

Some of this is available at

`https://dylanm.org/drafts/flexibly-graded-monads.pdf`