Flexible presentations of graded monads

Shin-ya Katsumata    Dylan McDermott
Tarmo Uustalu       Nicolas Wu
Motivation

1. Effects can be modelled using monads
2. which often come from presentations
3. which induce algebraic operations
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

   \[
   \begin{align*}
   \text{fail} & : 0 \quad \text{or} : 2 \\
   \text{or} (\text{fail}, x) &= x = \text{or} (x, \text{fail}) \\
   \text{or} (\text{or} (x, y), z) &= \text{or} (x, \text{or} (y, z))
   \end{align*}
   \]
3. which induces algebraic operations

   \[
   \begin{align*}
   \text{fail}_X &= (\lambda_\_ \cdot []) : 1 \to \text{List } X \\
   \text{or}_X &= (\lambda (xs, ys) . xs ++ ys) : \text{List } X \times \text{List } X \to \text{List } X
   \end{align*}
   \]
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

\[
\text{or}(\text{or}(\text{or}(\text{return11}, \text{return12}), \text{fail}), \\
\text{or}(\text{return13}, \text{cut})), \text{return14})
\]

is equivalent to

\[
\text{or}(\text{return11}, \\
\text{or}(\text{or}(\text{return12}, \text{or}(\text{return13}, \text{cut}))))
\]
Running example: nondeterminism with backtracking and cut

These computations can be modelled using a monad Cut

\[ \text{Cut } X = \text{List } X \times \{ \text{cut, nocut} \} \]

which has a presentation involving \text{or} : 2, \text{fail} : 0, \text{cut} : 0 \text{ [Piróg and Staton '17]}
Running example: nondeterminism with backtracking and cut

\[ \text{or}(t, u) \equiv t \quad \text{if } t \text{ cuts} \]
Assign grades $e \in \{\bot, 1, \top\}$ to computations:

\[
\begin{align*}
\top & \quad \text{don’t know anything} & t_1 \text{ has grade } e_1 & \quad \text{or (} t_1, t_2 \text{) has grade } (e_1 \cap e_2) \\
\forall & \quad \text{definitely cuts} & \quad \text{or returns something} & t_2 \text{ has grade } e_2 \\
1 & \quad \text{definitely cuts} & t \text{ has grade } e & e \leq e' \\
\forall & \quad \text{definitely cuts} & t \text{ has grade } e' & \text{fail has grade } \top \\
\bot & \quad \text{definitely cuts} & \text{cut has grade } \bot \\
\end{align*}
\]

Then:

\[\text{or}(t, u) \equiv t \quad \text{if } t \text{ has grade } \bot\]
Running example: nondeterminism with backtracking and cut

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

- $\top$: don’t know anything
- $\bot$: definitely cuts
- $1$: definitely cuts or returns something
- $\bot$: definitely cuts

Graded monad $\text{Cut}$:

$$\text{Cut}Xe = \{(xs, c) \in \text{List}X \times \{\text{cut}, \text{nocut}\} \mid (e = \bot \Rightarrow c = \text{cut}) \land (e = 1 \Rightarrow c = \text{cut} \lor xs \neq [])\}$$

Kleisli extension:

- $\top \cdot e = \top$
- $1 \cdot e = e$
- $\bot \cdot e = \bot$

Then:

$$f : X \to \text{Cut}Y e \quad f^{\dagger} : \text{Cut}Xd \to \text{Cut}Y(d \cdot e)$$
Rigidly graded presentations [Smirnov '08, Milius et al. '15, Dorsch et al. '19, Kura '20]

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

\[
\begin{align*}
    t_1 \text{ has grade } e & \quad \cdots \quad t_n \text{ has grade } e \\
    \text{op}(t_1, \ldots, t_n) \text{ has grade } d \cdot e
\end{align*}
\]
Rigidly graded presentations [Smirnov '08, Milius et al. '15, Dorsch et al. '19, Kura '20]

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

$t_1$ has grade $e$ \quad \cdots \quad t_n$ has grade $e$

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

These work well mathematically, but:

$t_1$ has grade $e_1$ \quad $t_2$ has grade $e_2$

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

For \text{or}, we must have $d \geq 1$, but then $\text{or}($cut, return 14$)$ will not have grade $\bot$
Flexibly graded presentations

\[ t_1 \text{ has grade } d'_1 \cdot e \quad \cdots \quad t_n \text{ has grade } d'_n \cdot e \]
\[ \text{op}(t_1, \ldots, t_n) \text{ has grade } d \cdot e \]

\[ t_1 \text{ has grade } e_1 \quad t_2 \text{ has grade } e_2 \]
\[ \text{or}(t_1, t_2) \text{ has grade } (e_1 \sqcap e_2) \]
Grading

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

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

Examples:

- Nondeterminism with cut: \((E, \leq) = \{ \bot \leq 1 \leq \top\}\)
  
  \[
  \begin{align*}
  T \cdot e & = T \\
  1 \cdot e & = e \\
  \bot \cdot e & = \bot
  \end{align*}
  \]

- Gifford-style effect systems: \((P\{\text{get, put, raise, ...}\}, \emptyset, \cup, \subseteq)\)
Flexibly graded presentations

Syntax:

- a **flexibly graded signature** is a collection of operations
- given a signature $\Sigma$, generate terms

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

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

\[
\Gamma \vdash t \equiv u : d
\]

Semantics $\sim$ graded monads
Terms and substitution

Terms in context:

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

Variables:

\[
\frac{x_1 : d'_1, \ldots, x_n : d'_n \vdash x_i : d'_i}{x_1 : d'_1, \ldots, x_n : d'_n \vdash t : d}
\]

Substitution:

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

A special case:

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

\[
f : [n] \rightarrow \text{Cut} Y e
\]

\[
f^\dagger_d : \text{Cut} [n] d \rightarrow \text{Cut} Y (d \cdot e)
\]
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

\[ \text{or}_{d_1, d_2} \in \Sigma(d_1, d_2; (d_1 \cap d_2)) \quad (\text{for each } d_1, d_2 \in \mathbb{E}) \]

\[ \text{fail} \in \Sigma(\cdot; \top) \]

\[ \text{cut} \in \Sigma(\cdot; \bot) \]
Terms

Given a signature $\Sigma$, generate terms by

$$
\begin{align*}
(x : d) &\in \Gamma & d \leq d' &\quad \Gamma \vdash t : d \\
\Gamma \vdash x : d & & \Gamma \vdash (d \leq d')^* t : d'
\end{align*}
$$

$$
\begin{align*}
\text{op} &\in \Sigma(d'_1, \ldots, d'_n; d) & \Gamma \vdash t_1 : d'_1 \cdot e & \cdots & \Gamma \vdash t_n : d'_n \cdot e \\
\Gamma \vdash \text{op}(e; t_1, \ldots, t_n) : d \cdot e
\end{align*}
$$

Substitution:

$$
\begin{align*}
(\text{op}(e; t_1, \ldots, t_n))\{e' ; x_1 \mapsto u_1, \ldots \}
\end{align*}
$$

$$
= \text{op}(e \cdot e' ; t_1\{e' ; x_1 \mapsto u_1, \ldots \}, \ldots, t_n\{e' ; x_1 \mapsto u_1, \ldots \})
$$
Terms

Given a signature $\Sigma$, generate terms by

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

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

Example

$$
\frac{\Gamma \vdash t_1 : d'_1 \cdot e \quad \Gamma \vdash t_2 : d'_2 \cdot e}{\Gamma \vdash \text{or}_{d'_1, d'_2}(e; t_1, t_2) : (d'_1 \sqcap d'_2) \cdot e \quad (= (d'_1 \cdot e) \sqcap (d'_2 \cdot e))}
\quad
\frac{(\text{or}_{d'_1, d'_2} \in \Sigma(d'_1, d'_2; (d'_1 \sqcap d'_2)))}{\Gamma \vdash \text{fail}(e; ) : \top \cdot e \quad (= \top)}
\quad
\frac{(\text{fail} \in \Sigma(\ ; \top))}{\Gamma \vdash \text{cut}(e; ) : \bot \cdot e \quad (= \bot)}
\quad
\frac{(\text{cut} \in \Sigma(\ ; \bot))}{\Gamma \vdash \text{cut}(e; ) : \bot \cdot e \quad (= \bot)}
$$
**Flexibly graded presentations**

**Definition**
A flexibly graded presentation consists of

- a signature \( \Sigma \)
- 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**

\[
\begin{align*}
x \colon e_1 \cdot d, y \colon e_2 \cdot d \vdash & \ or_{e_1,e_2}(d; x, y) \equiv or_{e_1 \cdot d,e_2 \cdot d}(1; x, y) : (e_1 \sqcap e_2) \cdot d \\
x \colon e_1, y \colon e_2 \vdash & \ (e_1 \sqcap e_2 \leq e_1' \sqcap e_2')^*(or_{e_1,e_2}(1; x, y)) \equiv or_{e_1,e_2}(1; (e_1 \leq e_1')^*x, (e_2 \leq e_2')^*y) : e_1' \sqcap e_2' \\
x \colon e \vdash & \ or_{\top,e}(1; \text{fail}(1;)), x) \equiv x : e \\
x \colon e \vdash x \equiv & \ or_{e,\top}(1; x, \text{fail}(1; )) : e \\
x \colon e_1, y \colon e_2, z : e_3 \vdash & \ or_{e_1 \sqcap e_2,e_3}(1; or_{e_1,e_2}(1; x, y), z) \equiv or_{e_1,e_2 \sqcap e_3}(1; x, or_{e_2,e_3}(1; y, z)) : e \\
x \colon \bot, y \colon e \vdash & \ or_{\bot,e}(1; x, y) \equiv x : \bot
\end{align*}
\]
Example: stacks of booleans

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

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

\[
\begin{align*}
\text{push}_v & \in \Sigma(0; 1) \\
\Gamma \vdash t : e & \quad \Gamma \vdash \text{push}_v(e; t) : 1 + e \quad (v \in \{\text{true}, \text{false}\}) \\
\Gamma \vdash t_{\text{empty}} : e & \quad \Gamma \vdash u_{\text{true}} : 1 + e \quad \Gamma \vdash u_{\text{false}} : 1 + e \\
\Gamma \vdash \text{pop}(e; t_{\text{empty}}, u_{\text{true}}, u_{\text{false}}) : e
\end{align*}
\]

- Equations:

\[
\begin{align*}
push_{\text{true}}(0; \text{pop}(0; x, y_{\text{true}}, y_{\text{false}})) & \equiv y_{\text{true}} \\
push_{\text{false}}(0; \text{pop}(0; x, y_{\text{true}}, y_{\text{false}})) & \equiv y_{\text{false}} \\
\text{pop}(0; x, push_{\text{true}}(0; x), push_{\text{false}}(0; x)) & \equiv x \\
\text{pop}(0; \text{pop}(0; x, y_{\text{true}}, y_{\text{false}}), z_{\text{true}}, z_{\text{false}}) & \equiv \text{pop}(0; x, z_{\text{true}}, z_{\text{false}})
\end{align*}
\]
Flexibly graded equational logic

Generate

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

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

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

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

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

A graded set \( X \) is a functor \( X : (E, \leq) \rightarrow \text{Set} \):
- a set \( X_e \) for each \( e \in E \) (elements of \( X \) of grade \( e \))
- a function \( (e \leq e')^* : X_e \rightarrow X_{e'} \) for each \( e \leq e' \in E \)

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

Example: for each presentation \( (\Sigma, E) \) and context \( \Gamma \)

\[
\text{Tm}_{(\Sigma, E)} \Gamma e = \{ [t] \equiv \mid \Gamma \vdash t : e \}
\]
Graded monads

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

A graded monad \( T \) (on \( \text{Set} \)) consists of:

- a graded set \( TX \) for each (ungraded) set \( X \)
- unit functions \( \eta_X : X \to TX1 \)
- Kleisli extension

\[
\begin{align*}
\eta_X & : X \to TX1 \\
f & : X \to TYe \\
\eta_X & : X \to TYe \\
\end{align*}
\]

natural in \( d, e \)

satisfying some laws

Example

Cut is a graded monad:

\[
\begin{align*}
\text{Cut } X e & = \{ (xs, c) \in \text{List} X \times \{ \text{cut, nocut} \} \} \\
& \mid (e = \bot \Rightarrow c = \text{cut}) \\
& \wedge (e = 1 \Rightarrow c = \text{cut} \lor xs \neq []) \\
\eta_X x & = ([x], \text{nocut}) \\
f_d^+( [x_1, \ldots, x_n], c) & = f x_1 \oplus \cdots \oplus f x_n \oplus ([], c) \\
(y, \text{cut}) \oplus (y', c) & = (y, \text{cut}) \\
(y, \text{nocut}) \oplus (y', c) & = (y + y', c)
\end{align*}
\]
Algebraic operations

Definition

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

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

natural in \(e\) and satisfying

\[
f_{d \cdot e}^\dagger (\alpha_{X,e}(t_1, \ldots, t_n)) = \alpha_{Y,e' \cdot e}(f_{d'_1 \cdot e t_1, \ldots, f_{d'_n \cdot e t_n}}) \quad (f : X \to TYe')
\]

Example

For the graded monad \(\text{Cut}\), we have

\[
\begin{align*}
\llbracket \text{or}_{d'_1,d'_2} \rrbracket_{X,e} &= (\oplus) : \text{Cut}X(d'_1 \cdot e) \times \text{Cut}X(d'_2 \cdot e) \to \text{Cut}X((d'_1 \sqcap d'_2) \cdot e) \\
\llbracket \text{fail} \rrbracket_{X,e} &= (\lambda_\_ \cdot ([], \text{nocut})) : 1 \to \text{Cut}X(\top \cdot e) \\
\llbracket \text{cut} \rrbracket_{X,e} &= (\lambda_\_ \cdot ([], \text{cut})) : 1 \to \text{Cut}X(\bot \cdot e)
\end{align*}
\]
Presenting graded monads

Given a flexibly graded presentation \((\Sigma, E)\), we want

- a graded monad \(T_{(\Sigma,E)}\)
- with a \((d_1', \ldots, d_n'; d)\)-ary algebraic operation

\[
\left[ \text{op} \right]_{X,e} : \prod_i T_{(\Sigma,E)} X(d_i' \cdot e) \rightarrow T_{(\Sigma,E)} X(d \cdot e)
\]

for each \(\text{op} \in \Sigma(d_1', \ldots, d_n'; d)\) (satisfying equations)

- that is in some sense canonical
Algebras

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

- a graded set \(A\)
- with a natural family of functions
  
  \[
  \llbracket \text{op} \rrbracket_e : \prod_i A(d'_i \cdot e) \rightarrow A(d \cdot e)
  \]

  for each \(\text{op} \in \Sigma(d'_1, \ldots, d'_n; d)\)
- such that
  
  \[
  \llbracket t \rrbracket_e = \llbracket u \rrbracket_e : \prod_i A(d'_i \cdot e) \rightarrow A(d \cdot e)
  \]

  for each \(e \in E\) and axiom \(x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d\)

Example

- \(T_{(\Sigma, E)} X\), with algebraic operations \(\llbracket \text{op} \rrbracket_X\)
- \(\text{Tm}_{(\Sigma, E)} \Gamma\), with \(\llbracket \text{op} \rrbracket_e ([t_1]_\equiv, \ldots, [t_n]_\equiv) = [\text{op}(e; t_1, \ldots, t_n)]_\equiv\)

The equational logic is sound and complete:

\[
\Gamma \vdash t \equiv u : d \quad \iff \quad \text{for all } (\Sigma, E)\text{-algebras } (A, [-]), \quad \llbracket t \rrbracket = \llbracket u \rrbracket
\]
Algebras

If \((\Sigma, E)\) is a flexibly graded presentation, a \((\Sigma, E)\)-algebra \((A, \llbracket - \rrbracket)\) is

- a graded set \(A\)
- with a natural family of functions

\[
\llbracket \text{op} \rrbracket_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)
\]

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

\[
\llbracket t \rrbracket_e = \llbracket u \rrbracket_e : \prod_i A(d'_i \cdot e) \to A(d \cdot e)
\]

for each \(e \in E\) and axiom \(x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d\)

A morphism \(f : (A, \llbracket - \rrbracket) \to (A', \llbracket - \rrbracket')\) is a natural family of functions

\[
f_d : A d \to A' d
\]

preserving \(\llbracket \text{op} \rrbracket\)
Algebras

If \((\Sigma, E)\) is a flexibly graded presentation, a \((\Sigma, E)\)-algebra \((A, \llbracket - \rrbracket)\) is

- a graded set \(A\)
- with a natural family of functions

\[ \llbracket \text{op} \rrbracket_e : \prod_i A(d'_i \cdot e) \rightarrow A(d \cdot e) \]

for each \(\text{op} \in \Sigma(d'_1, \ldots, d'_n; d)\)

- such that

\[ \llbracket t \rrbracket_e = \llbracket u \rrbracket_e : \prod_i A(d'_i \cdot e) \rightarrow A(d \cdot e) \]

for each \(e \in E\) and axiom \(x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d\)

A morphism \(f : (A, \llbracket - \rrbracket) \rightarrow (A', \llbracket - \rrbracket')\) of grade \(e\) is a natural family of functions

\[ f_d : Ad \rightarrow A'(d \cdot e) \]

preserving \(\llbracket \text{op} \rrbracket\)
Locally graded categories [Wood ’76]

Definition
A *locally graded category* $C$ consists of
- a collection $\vert C \vert$ of objects
- graded sets $C(X, Y)$ of morphisms ($f : X \rightarrow Y$ means $f \in C(X, Y)e$)
- identities $\text{id}_X : X \rightarrow X$
- composition

\[
\frac{f : X \rightarrow Y \quad g : Y \rightarrow Z}{g \circ f : X \rightarrow Z}
\]

natural in $e, e'$
such that

\[
\text{id}_Y \circ f = f = f \circ \text{id}_X \quad (h \circ g) \circ f = h \circ (g \circ f)
\]

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

Every graded monad $T$ has a locally graded Kleisli category $\text{Kl}(T)$:
- Objects are sets $X$
- Morphisms $f : X \to Y$ are functions $f : X \to TYe$

The locally graded category $\text{GSet}$:
- Objects are graded sets
- Morphisms $f : X \to Y$ 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

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

$\text{Alg}(\Sigma, E)$:
- Objects are $(\Sigma, E)$-algebras
- Morphisms are as in $\text{GSet}$, but preserving $\lbrack - \rbrack$
- Identities and composition: as in $\text{GSet}$
Functors

Definition
A functor $F : C \to D$ between locally graded categories is an object mapping $F : |C| \to |D|$ with a mapping of morphisms

$$f : X \rightarrowrightarrow Y$$

$$Ff : FX \rightarrowrightarrow FY$$

natural in $e$, and preserving identities and composition.

There is a forgetful functor

$$U_{(\Sigma,E)} : \text{Alg}(\Sigma,E) \to \text{GSet}$$

$$(A, \llbracket - \rrbracket) \mapsto A$$

$$f \mapsto f$$
Algebra

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

- a graded set $A$
- with an extension operator

\[
\begin{align*}
  f &: X \rightarrow Ae \\
  f^+_d &: TXd \rightarrow A(d \cdot e)
\end{align*}
\]

- satisfying some laws

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

\[ U_T : EM(T) \rightarrow GSet \]
Presenting graded monads

**Theorem**

For every flexibly graded presentation $(\Sigma, E)$, there is

- a graded monad $T_{(\Sigma, E)}$
- and functor $R_{(\Sigma, E)} : \text{Alg}(\Sigma, E) \to \text{EM}(T_{(\Sigma, E)})$ over $\text{GSet}$

such that

- $(T_{(\Sigma, E)}X, (-)^\dagger) = R_{(\Sigma, E)}(T_{(\Sigma, E)}X, \llbracket - \rrbracket_X)$ for some $\llbracket - \rrbracket_X$
- for every graded monad $T'$ and functor $R' : \text{Alg}(\Sigma, E) \to \text{EM}(T')$ over $\text{GSet}$, there is a unique $F : \text{EM}(T_{(\Sigma, E)}) \to \text{EM}(T')$ over $\text{GSet}$ such that

$$
\begin{array}{c}
\text{Alg}(\Sigma, E) \xrightarrow{R_{(\Sigma, E)}} \text{EM}(T_{(\Sigma, E)}) \\
\downarrow R' \quad \downarrow F \\
\text{EM}(T')
\end{array}
$$
Presenting graded monads

For the presentation of nondeterminism with Cut

\[ T_{(\Sigma,E)} \cong \text{Cut} \]

with algebraic operations \([\text{cut}_{d_1,d_2}'], [\text{fail}], [\text{cut}]\)

For the presentation of stacks of booleans:

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

\[ [\text{push}_v]_{X,e} : \text{Stk}Xe \to \text{Stk}X(1 + e) \]

\[ [\text{push}_v]_{X,e} \ t \ \text{vs} = t(v :: \text{vs}) \]

\[ [\text{pop}]_{X,e} : \text{Stk}Xe \times \text{Stk}X(1 + e) \times \text{Stk}X(1 + e) \to \text{Stk}Xe \]

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

- **Flexibly graded presentations**
  
  $$(\Sigma,E) \leftrightarrow Tm_{(\Sigma,E)}$$

- **Flexibly graded clones**
  
  = sets of terms, with variables and substitution

- **Flexibly graded monads**
  
  = monad on $GSet$

- **Composed with**
  
  left Kan extension along $FCtx \rightarrow GSet$

- **Graded monads**
  
  compose with $Free(Set) \rightarrow GSet$
Constructing $T(\Sigma, E)$

flexibly graded presentations

$\Sigma, E \mapsto Tm_{(\Sigma, E)}$ \cong flexibly graded clones

left Kan extension along $\text{FCtx} \to \text{GSet}$ \cong compose with $\text{FCtx} \to \text{GSet}$

flexibly graded monads

preserving conical sifted colimits

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

graded monads

preserving conical sifted colimits

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

$= (\text{FCtx} \to \text{GSet})$-relative monad

$= \text{monad on GSet}$

$= \text{preserving conical sifted colimits}$

$= (\text{Free}(\text{Set}) \to \text{GSet})$-relative monad

$= \text{preserving conical sifted colimits}$
Given a flexibly graded presentation \((\Sigma, E)\), there is

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

\[
\llbracket \text{op} \rrbracket_{X,e} : \prod_i T_{(\Sigma, E)}X(d'_i \cdot e) \rightarrow T_{(\Sigma, E)}X(d \cdot e)
\]

for each \(\text{op} \in \Sigma(d'_1, \ldots, d'_n; d)\) (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