Flexible presentations of graded monads

Shin-ya Katsumata\textsuperscript{1}  Dylan McDermott\textsuperscript{2}  Tarmo Uustalu\textsuperscript{2,3}  Nicolas Wu\textsuperscript{4}

\textsuperscript{1} National Institute of Informatics, Japan
\textsuperscript{2} Reykjavik University, Iceland
\textsuperscript{3} Tallinn University of Technology, Estonia
\textsuperscript{4} Imperial College London, UK
Example: nondeterminism with backtracking and cut

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

Terms should satisfy some equations:

\[
\text{or}(M, N) \equiv M \quad \text{whenever } M \text{ cuts}
\]

\[
\left( \begin{array}{c}
do x \leftarrow N_1 \\
y \leftarrow N_2 \\
M
\end{array} \right) \equiv \left( \begin{array}{c}
do y \leftarrow N_2 \\
x \leftarrow N_1 \\
M
\end{array} \right) \quad \text{whenever } M \text{ cuts}
\]

\[
\left( \begin{array}{c}
do x \leftarrow N_1 \\
y \leftarrow N_2 \\
M
\end{array} \right) \equiv \left( \begin{array}{c}
do y \leftarrow N_2 \\
x \leftarrow N_1 \\
M
\end{array} \right) \quad \text{whenever } M \text{ cuts}
\]

\[
\]

and \( N_1 \) cuts or returns something

and \( N_2 \) cuts or returns something
Models of effects from presentations

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: (based on [Piróg and Staton '17])

1. Nondeterminism with can be modelled using a monad Cut

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

2. which comes from the presentation of monoids with a left zero:

\[ \text{or} : 2 \quad \text{fail} : 0 \quad \text{cut} : 0 \]

\[ \text{or}(\text{or}(x, y), z) \equiv \text{or}(x, \text{or}(y, z)) \quad \text{or}((\text{fail}, x)) \equiv x \equiv \text{or}(x, \text{fail}) \quad \text{or}(\text{cut}, x) \equiv x \]

3. which induces algebraic operations

\[ \text{or}_X : \text{Cut}_X \times \text{Cut}_X \to \text{Cut}_X \]

\[ \text{fail}_X : 1 \to \text{Cut}_X \quad \text{cut}_X : 1 \to \text{Cut}_X \]
Example: nondeterminism with backtracking and cut

\[
\text{or}(M, N) \equiv M \quad \text{whenever } M \text{ cuts}
\]

\[
\left( \begin{array}{c}
\text{do } x \leftarrow N_1 \\
y \leftarrow N_2 \\
M
\end{array} \right) \equiv
\left( \begin{array}{c}
\text{do } y \leftarrow N_2 \\
x \leftarrow N_1 \\
M
\end{array} \right) \quad \text{whenever } M \text{ cuts}
\]

and \( N_1 \) cuts or returns something

and \( N_2 \) cuts or returns something
Grading

Grade computations by elements of an ordered monoid:

$$(\mathbb{E}, \leq, 1, \cdot)$$

so that they form a graded monad

For example:

- lists graded by $(\mathbb{N}, =, 1, \cdot)$
  $$\text{Vec} X e = \text{lists over } X \text{ of length exactly } e$$

- lists graded by $(\mathbb{N}, \leq, 1, \cdot)$
  $$\text{BVec} X e = \text{lists over } X \text{ of length at most } e$$
Example: grading nondeterminism with backtracking and cut

or\((M,N) \equiv M\) whenever \(M\) has grade \(\perp\)

Assign grades \(e \in \{\perp, 1, \top\}\) to computations:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\top)</td>
<td>don’t know anything</td>
</tr>
<tr>
<td>(\bot)</td>
<td>definitely cuts</td>
</tr>
<tr>
<td>(1)</td>
<td>definitely cuts</td>
</tr>
<tr>
<td>(\bot)</td>
<td>definitely cuts</td>
</tr>
</tbody>
</table>

Graded monad Cut:

\[
\text{Cut}_Xe = \{ (xs, c) \in \text{List}X \times \{\text{cut, nocut}\} \mid (e = \perp \Rightarrow c = \text{cut}) \\
\quad \wedge (e = 1 \Rightarrow c = \text{cut} \vee xs \neq []) \}
\]

Kleisli extension:

\[
f : X \to \text{Cut}_Ye \\
\overset{\dagger}{f_d} : \text{Cut}_Xd \to \text{Cut}_Y(d \cdot e)
\]

where \(\top \cdot e = \top\), \(1 \cdot e = e\), \(\bot \cdot e = \bot\)
Example: grading nondeterminism with backtracking and cut

1. Nondeterminism with cut can be modelled using a graded monad $\text{Cut}$

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

2. which comes from a graded presentation of monoids with a left zero?
3. which induces graded algebraic operations?

$$\text{or}_{d_1, d_2, X} : \text{Cut} X d_1 \times \text{Cut} X d_2 \rightarrow \text{Cut} X (d_1 \sqcap d_2) \quad (d_1, d_2 \in \{\bot, 1, \top\})$$

$$\text{fail}_X : 1 \rightarrow \text{Cut} X \top \quad \text{cut}_X : 1 \rightarrow \text{Cut} X \bot$$

The existing notions of graded presentation are not general enough

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

Develop a notion of **flexibly graded presentation**

- Every flexibly graded presentation \((\Sigma, E)\) induces
  - a canonical graded monad \(T_{(\Sigma,E)}\)
  - along with a **flexibly graded algebraic operation** for each operation of the presentation

- Examples like Cut have computationally natural flexibly graded presentations

- The constructions are mathematically justified by locally graded categories, and a notion of **flexibly graded abstract clone**
Flexibly graded presentations

Given an ordered monoid \((\mathbb{E}, \leq, 1, \cdot)\) of grades, a **flexibly graded presentation** \((\Sigma, \mathbb{E})\) consists of

- a signature \(\Sigma\): sets
  \[
  \Sigma(d'_1, \ldots, d'_n; d)
  \]
  of operations
  \[
  e \in \mathbb{E} \quad \Gamma \vdash t_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash t_n : d'_n \cdot e
  \]
  \[
  \Gamma \vdash \text{op}(e; t_1, \ldots, t_n) : d \cdot e
  \]

- a collection of axioms \(\mathbb{E}\): sets
  \[
  \mathbb{E}(d'_1, \ldots, d'_n; d)
  \]
  of equations
  \[
  x_1 : d'_1, \ldots, x_n : d'_n \vdash t \equiv u : d
  \]

Part of the presentation of nondeterminism with cut:

- grades \(\mathbb{E} = \{\bot \leq 1 \leq \top\}\)
  
  \[
  \Gamma \vdash t_1 : d'_1 \cdot e \quad \Gamma \vdash t_2 : d'_2 \cdot e
  \]
  
  \[
  \Gamma \vdash \text{or}_{d'_1, d'_2}(e; t_1, t_2) : (d'_1 \sqcap d'_2) \cdot e
  \]
  
  or \(\bot, e(1; x, y) \equiv x\)
Semantics

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

- a notion of \((\Sigma, E)\)-algebra, forming a **locally graded category** \(\text{Alg}(\Sigma, E)\)

\[\text{A } (\Sigma, E)\text{-algebra } (A, \llbracket - \rrbracket) \text{ is:} \]

- a graded set \(A : B \rightarrow \text{Set}\)
- with an interpretation

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

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

- satisfying each axiom \(t \equiv u\) of \(E\):

\[\llbracket t \rrbracket_e = \llbracket u \rrbracket_e \text{ for every } e\]
Semantics

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

- a notion of \((\Sigma, E)\)-algebra, forming a locally graded category \(\text{Alg}(\Sigma, E)\) [Wood '76]
- a sound and complete equational logic

\[
\Gamma \vdash t \equiv u : d \quad \text{generated by}
\]

\[
(t, u) \in E(d'_1, \ldots, d'_n; d) \quad \Gamma \vdash s_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash s_n : d'_n \cdot e
\]

\[
\Gamma \vdash t\{e; x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\} \equiv u\{e; x_1 \mapsto s_1, \ldots, x_n \mapsto s_n\} : d \cdot e
\]

and some other rules

Soundness and completeness:

\[
\llbracket t \rrbracket = \llbracket u \rrbracket \quad \text{in every} \ (\Sigma, E)\text{-algebra} \quad \iff \quad \Gamma \vdash t \equiv u : d \text{ is derivable}
\]
Semantics

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

- a notion of \((\Sigma, E)\)-algebra, forming a locally graded category \(\text{Alg}(\Sigma, E)\) [Wood ’76]
- a sound and complete equational logic
- a graded monad \(T_{(\Sigma,E)}\) on \(\text{Set}\) and concrete functor \(R_{(\Sigma,E)} : \text{Alg}(\Sigma, E) \to \text{EM}(T_{(\Sigma,E)})\), satisfying a universal property

For every graded monad \(T'\) and concrete functor \(R' : \text{Alg}(\Sigma, E) \to \text{EM}(T')\): 

\[
\begin{array}{ccc}
\text{Alg}(\Sigma, E) & \xrightarrow{R_{(\Sigma,E)}} & \text{EM}(T_{(\Sigma,E)}) \\
\downarrow R' & & \downarrow \text{EM}(\alpha) \\
\text{EM}(T') & \xrightarrow{\alpha} & T'
\end{array}
\]

(But \(R_{(\Sigma,E)}\) is usually not an isomorphism)
Semantics

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

- a notion of \((\Sigma, E)\)-algebra, forming a locally graded category \(\text{Alg}(\Sigma, E)\)
- a sound and complete equational logic
- a graded monad \(T_{(\Sigma,E)}\) on \(\text{Set}\) and concrete functor
  \(R_{(\Sigma,E)} : \text{Alg}(\Sigma, E) \rightarrow \text{EM}(T_{(\Sigma,E)})\), satisfying a universal property
- for every op in \(\Sigma\), a flexibly graded algebraic operation for \(T_{(\Sigma,E)}\)

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

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

natural in \(e\), and compatible with Kleisli extension

(Because each free \(T_{(\Sigma,E)}\)-algebra \(T_{(\Sigma,E)}X\) forms a \((\Sigma, E)\)-algebra)
For every flexibly graded presentation \((\Sigma, E)\), there is:

- a notion of \((\Sigma, E)\)-algebra, forming a \textit{locally graded category} \(\text{Alg}(\Sigma, E)\) \cite{Wood '76}
- a sound and complete equational logic
- a graded monad \(T_{(\Sigma, E)}\) on \(\text{Set}\) and concrete functor
  \(R_{(\Sigma, E)} : \text{Alg}(\Sigma, E) \to \text{EM}(T_{(\Sigma, E)})\), satisfying a universal property
- for every op in \(\Sigma\), a flexibly graded algebraic operation for \(T_{(\Sigma, E)}\)

A large class of graded monads have flexibly graded presentations:

- exactly the finitary graded monads on \(\text{Set}\)
- correspondence goes via flexibly graded clones

Graded monads we care about have natural flexibly graded presentations