Flexible presentations of graded monads

Shin-ya Katsumata\(^1\), Dylan McDermott\(^2\), Tarmo Uustalu\(^{2,3}\), and Nicolas Wu\(^4\)

\(^1\) National Institute of Informatics, Japan s-katsumata@nii.ac.jp
\(^2\) Reykjavik University, Iceland dylans@ru.is,tarmo@ru.is
\(^3\) Tallinn University of Technology, Estonia
\(^4\) Imperial College London, UK n.wu@imperial.ac.uk

Consider a language in which we can express backtracking computations using an operation or for nondeterministic choice, and an operation cut for pruning any remaining choices. Let \(t\) be the computation \(\text{or}(\text{return } 17, \text{cut})\), which offers only 17 as a possible result, and prunes the rest of the search space. The computation \(\text{or}(t, \text{return } 42)\) is equivalent to \(t\), and more generally, the equation \(\text{or}(x, y) \approx x\) is valid whenever we know that \(x\) definitely cuts. We may seek to analyse computations statically to determine whether they cut, and whether we can therefore simplify a program using \(\text{or}(x, y) \approx x\). One approach to doing this is through grading. We assign a grade \(\perp\) to each computation we know will cut, and propagate this information throughout the program (other computations get other grades). This approach has a well-established semantics using \textit{graded monads} [9, 5, 2]. There is a graded monad \textbf{Cut} that models our backtracking example; it is similar to Piróg and Staton’s non-graded monad [7]. Piróg and Staton show that their monad has a \textit{presentation} in terms of operations for nondeterministic choice and cut. We may expect there to be a similar presentation of \textbf{Cut}, using the existing notions of graded presentation [9, 6, 1, 3], which we call \textit{rigidly graded presentations}. However, rigidly graded presentations have a deficiency: they only allow operations to be applied when all arguments have the same grade. Above \(t\) has grade \(\perp\) because one argument to \text{cut} has grade \(\perp\), but the other does not. A rigidly graded presentation would assign some grade to \(t\), by overapproximating, but not \(\perp\), so the analysis would be imprecise. This is a problem in other applications, such as: mutable state graded by relations (relating initial states to final states); stack-based computations graded by bounds on the change in stack height; and nondeterministic computations graded by upper bounds on the number of options that are chosen from.

While rigidly graded presentations are motivated by their theory (which includes a correspondence with a class of graded monads, analogous to the classical monad–algebraic theory correspondence), they are unsuitable when it comes to applications. We introduce a more general notion of \textit{flexibly graded} presentation that does not suffer from the same issue.

\textbf{Grading} We recall the notion of graded monad (on \textit{Set}). The \textit{grades} are elements of an ordered monoid \(((\mathbb{E}), \leq, 1, \cdot)\). A grade \(e \in \mathbb{E}\) abstractly quantifies the effect of a computation; the order \(\leq\) provides overapproximation of grades, the unit 1 is the grade of a trivial computation, and the multiplication \(\cdot\) provides the grade of a sequence of two computations. For the backtracking example above the poset \(((\mathbb{E}), \leq)\) is \(\{\perp \leq 1 \leq \top\}\), where \(\perp\) means ‘definitely cuts’, the unit grade 1 means ‘definitely either cuts or produces at least one value’, and \(\top\) imposes no restrictions. Multiplication is given by \(\perp \cdot e = \perp, 1 \cdot e = e\) and \(\top \cdot e = \top\).

A \textit{graded set} \(Y\) is a family of sets \(Ye\), together with a \textit{coercion} function \((e \leq e')^* : Ye \rightarrow Ye'\) for each \(e \leq e'\), satisfying two equational conditions. A \textit{graded monad} \(R\) consists of a graded set \(RX\) and unit function \(\eta_X : X \rightarrow RX1\) for each set \(X\), and a Kleisli extension operation that maps functions \(f : X \rightarrow RYe\) and grades \(d\) to functions \(f^d : RXd \rightarrow RY(d \cdot e)\), satisfying some conditions. For \textbf{Cut}, computations over \(X\) of grade \(e\) are elements of the following set \(\text{Cut}Xe\), where \(c\) indicates whether the computation cuts (\(\perp\) for ‘cuts’, \(\top\) for ‘does not cut’).

\[
\text{Cut}Xe = \{(\vec{x}, c) \in \text{List}X \times \{\perp, \top\} \mid (e = \perp \Rightarrow c = \perp) \wedge (e = 1 \Rightarrow c = \perp \lor \vec{x} \neq [])\}
\]
Flexible graded presentations

In general, a presentation \((\Sigma, E)\) consists of a signature \(\Sigma\), specifying the operations and inducing a notion of term, and a set \(E\) of equational axioms, inducing an equational theory.

A flexibly graded signature \(\Sigma\) consists of a set \(\Sigma(d_I; d)\) of \((d_I; d)\)-ary operations for each list of grades \(d_I\) and grade \(d\). (Rigidly graded signatures correspond to the special case in which every operation has \(d_I = [1, \ldots, 1]\).) The terms over \(\Sigma\) are generated by the following rules for variables, coercions, and application of operations \(op \in \Sigma(d_I; d)\), where \(\Gamma = x_1 : d_{1}'_1, \ldots, x_m : d_{m}'_m\).

\[
\begin{align*}
1 \leq i \leq m & \quad \Gamma \vdash t_i : e_i \quad e_i \leq e'_i \\
\Gamma \vdash \frac{t}{e} & \quad \frac{\Gamma \vdash u_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash u_n : d'_n \cdot e}{\Gamma \vdash \text{op}(e; u_1, \ldots, u_n) : d \cdot e}
\end{align*}
\]

The grade \(e\) in the \(\text{op}\) rule has a crucial role: it is there precisely because of the grade \(e\) in the Kleisli extension above. Unlike in a rigidly graded presentation, variables can have different grades \(d_I\). In a flexibly graded presentation \((\Sigma, E)\), an equational axiom in \(E\) is a pair \((t, u)\) of terms of some grade \(e\) in some context \(\Gamma\). These axioms induce a notion of equality \(\Gamma \vdash t \approx u : e\). For the backtracking example, we have a flexibly graded version of Piróg and Staton’s non-graded presentation [7]. The signature has operations \(\text{cut}, \text{fail}, \text{or}_{d_1, d_2}\), giving rise to the following rules for constructing terms (where \(\cap\) denotes meet).

\[
\begin{align*}
\Gamma \vdash \text{cut}(e; t) : \bot & \quad \Gamma \vdash \text{fail}(e; t) : \top \\
\Gamma \vdash u_1 : d_1 \cdot e & \quad \Gamma \vdash u_2 : d_2 \cdot e \\
\Gamma \vdash \text{or}_{d_1, d_2}(e; u_1, u_2) : (d_1 \cap d_2) \cdot e
\end{align*}
\]

One of the axioms (we omit the rest) is \(x : \bot, y : 1 \vdash \text{or}_{1, 1}(1; x, y) \approx x : \bot\), which is the example we use in the introduction. This can be applied only when \(x\) has grade \(\bot\); such a restriction on the grade of a variable is not possible in a rigidly graded presentation.

**Semantics**

In classical universal algebra each presentation gives rise to a notion of algebra (a.k.a. model), consisting of a set with interpretations for the operations, validating the equations. The equational theory is sound and complete w.r.t. this notion of model. If \((\Sigma, E)\) is a flexibly graded presentation, a \(\Sigma\)-algebra \((E, \cdot)\) is a graded set \(A\) equipped with a natural transformation \([\text{op}] : \prod_{\Sigma} A(d'_i, \cdot) \Rightarrow A(d, \cdot)\) for each \(op \in \Sigma(d_I; d)\). These extend to interpretations \([t] : \prod_{\Sigma} A(d'_i, \cdot) \Rightarrow A(d, \cdot)\) of terms \(t_1 : d_1', \ldots, t_m : d_m' \vdash t : d\). A \(\Sigma\)-algebra is a \((\Sigma, E)\)-algebra when \([t] = [u]\) for each axiom \((t, u)\). The equational logic is sound and complete: an equation \(\Gamma \vdash t \approx u : e\) is derivable exactly when \([t] = [u]\) in every \((\Sigma, E)\)-algebra.

**Presenting graded monads**

In the classical correspondence between presentations and monads, the monad \(T^{(\Sigma, E)}\) induced by a presentation is completely determined by the fact that \(T^{(\Sigma, E)}\)-algebras are equivalently \((\Sigma, E)\)-algebras. For flexibly graded presentations the situation is more complex. In general, there is no graded monad whose algebras are \((\Sigma, E)\)-algebras, and we do not get a correspondence with graded monads. However, every flexibly graded presentation does induce a canonical graded monad \(R^{(\Sigma, E)}\). Every \((\Sigma, E)\)-algebra induces an \(R^{(\Sigma, E)}\)-algebra, and \(R^{(\Sigma, E)}\) is in some sense the universal graded monad with this property (we omit the precise statement). Moreover, free \(R^{(\Sigma, E)}\)-algebras form \((\Sigma, E)\)-algebras, so in particular the graded sets \(R^{(\Sigma, E)} X\) admit interpretations of the operations of \(\Sigma\). These interpretations form flexibly graded algebraic operations for \(R^{(\Sigma, E)}\) (which are analogous to algebraic operations for non-graded monads [8]). In this sense, \((\Sigma, E)\) does indeed present a graded monad \(R^{(\Sigma, E)}\).

The proof of this involves a notion of flexibly graded monad, introduced in [4]. There is an algebra-preserving correspondence between flexibly graded presentations and flexibly graded monads that preserve conical sifted colimits, and every flexibly graded monad induces a canonical (rigidly) graded monad [4, Section 5]. The latter is \(R^{(\Sigma, E)}\) if we start with \((\Sigma, E)\). Moreover, every graded monad \(R\) that preserves sifted colimits has a flexibly graded presentation.
References


