# Flexible presentations of graded monads

Shin-ya Katsumata[1], Dylan McDermott[2], Tarmo Uustalu[23], and Nicolas Wu[4]

[1] National Institute of Informatics, Japan s-katsumata@nii.ac.jp
[2] Reykjavik University, Iceland dylanm@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 $\texttt{or}(\texttt{return}\ 17, \texttt{cut})$, which offers only 17 as a possible result, and prunes the rest of the search space. The computation $\texttt{or}(t, \texttt{return}\ 42)$ is equivalent to $t$, and more generally, the equation $\texttt{or}(x, y) \approx x$ is valid whenever we know that $x$ definitely `cut`s. We may seek to analyse computations statically to determine whether they `cut`, and whether we can therefore simplify a program using $\texttt{or}(x, y) \approx x$. One approach to doing this is through *grading*. We assign a grade $\bot$ 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 *graded monads* [9, 5, 2]. There is a graded monad $\mathsf{Cut}$ 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 *presentation* in terms of operations for nondeterministic choice and cut. We may expect there to be a similar presentation of $\mathsf{Cut}$, using the existing notions of graded presentation [9, 6, 1, 3], which we call *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 $\bot$ because one argument to `cut` has grade $\bot$, but the other does not. A rigidly graded presentation would assign *some* grade to $t$, by overapproximating, but not $\bot$, 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 *flexibly graded* presentation that does not suffer from the same issue.

**Grading** We recall the notion of graded monad (on **Set**). The *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 $\{\bot \leq 1 \leq \top\}$, where $\bot$ 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 $\bot \cdot e = \bot$, $1 \cdot e = e$ and $\top \cdot e = \top$.

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

$$\mathrm{Cut}Xe = \{(\vec{x}, c) \in \mathrm{List}X \times \{\bot, \top\} \mid (e = \bot \Rightarrow c = \bot) \wedge (e = 1 \Rightarrow c = \bot \vee \vec{x} \neq [])\}$$

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

$$\frac{1 \leq i \leq m}{\Gamma \vdash x_i : d'_i} \qquad \frac{\Gamma \vdash t : e \quad e \leq e'}{\Gamma \vdash (e{\leq}e')^* t : e'} \qquad \frac{\Gamma \vdash u_1 : d'_1 \cdot e \quad \cdots \quad \Gamma \vdash u_n : d'_n \cdot e}{\Gamma \vdash \mathsf{op}(e; u_1, \ldots, u_n) : d \cdot e}$$

The grade $e$ in the 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 $\mathsf{cut}$, $\mathsf{fail}$, $\mathsf{or}_{d_1,d_2}$, giving rise to the following rules for constructing terms (where $\sqcap$ denotes meet).

$$\frac{}{\Gamma \vdash \mathsf{cut}(e;\ ) : \bot} \qquad \frac{}{\Gamma \vdash \mathsf{fail}(e;\ ) : \top} \qquad \frac{\Gamma \vdash u_1 : d_1 \cdot e \qquad \Gamma \vdash u_2 : d_2 \cdot e}{\Gamma \vdash \mathsf{or}_{d_1,d_2}(e; u_1, u_2) : (d_1 \sqcap d_2) \cdot e}$$

One of the axioms (we omit the rest) is $x : \bot, y : 1 \vdash \mathsf{or}_{\bot,1}(1; x, y) \approx x : \bot$, 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* is a graded set $A$ equipped with a natural transformation $[\![\mathrm{op}]\!] : \prod_i A(d'_i \cdot -) \Rightarrow A(d \cdot -)$ for each $\mathrm{op} \in \Sigma(\vec{d'}, d)$. These extend to interpretations $[\![t]\!] : \prod_i A(d'_i \cdot -) \Rightarrow A(d \cdot -)$ of terms $x_1 : d'_1, \ldots, x_n : d'_n \vdash t : d$. 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 $\mathsf{T}^{(\Sigma,E)}$ induced by a presentation is completely determined by the fact that $\mathsf{T}^{(\Sigma,E)}$-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 $\mathsf{R}^{(\Sigma,E)}$. Every $(\Sigma, E)$-algebra induces an $\mathsf{R}^{(\Sigma,E)}$-algebra, and $\mathsf{R}^{(\Sigma,E)}$ is in some sense the universal graded monad with this property (we omit the precise statement). Moreover, *free* $\mathsf{R}^{(\Sigma,E)}$-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 $\mathsf{R}^{(\Sigma,E)}$ (which are analogous to algebraic operations for non-graded monads [8]). In this sense, $(\Sigma, E)$ does indeed present a graded monad $\mathsf{R}^{(\Sigma,E)}$.

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 $\mathsf{R}^{(\Sigma,E)}$ if we start with $(\Sigma, E)$. Moreover, every graded monad $\mathsf{R}$ that preserves sifted colimits has a flexibly graded presentation.

# References

[1] Ulrich Dorsch, Stefan Milius, and Lutz Schröder. Graded monads and graded logics for the linear time–branching time spectrum. In Wan Fokkink and Rob van Glabbeek, editors, *30th Int. Conf. on Concurrency Theory, CONCUR 2019*, volume 140 of *Leibniz Int. Proc. in Informatics*, pages 36:1–36:16. Dagstuhl Publishing, Saarbrücken/Wadern, 2019.

[2] Shin-ya Katsumata. Parametric effect monads and semantics of effect systems. In *Proc. of 41st Ann. ACM SIGPLAN-SIGACT Symp. on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014*, pages 633–645. ACM Press, New York, 2014.

[3] Satoshi Kura. Graded algebraic theories. In Jean Goubault-Larrecq and Barbara König, editors, *Foundations of Software Science and Computation Structures: 23rd Int. Conf., FOSSACS 2020, Dublin, Ireland, April 25–30, 2020, Proceedings*, volume 12077 of *Lect. Notes in Comput. Sci.*, pages 401–421. Springer, Cham, 2020.

[4] Dylan McDermott and Tarmo Uustalu. Flexibly graded monads and graded algebras. Manuscript, available at https://dylanm.org/drafts/flexibly-graded-monads.pdf, 2022.

[5] Paul-André Melliès. Parametric monads and enriched adjunctions. Manuscript, 2012.

[6] Stefan Milius, Dirk Pattinson, and Lutz Schröder. Generic trace semantics and graded monads. In Lawrence S. Moss and Paweł Sobociński, editors, *6th Conf. on Algebra and Coalgebra in Computer Science, CALCO 2015*, volume 35 of *Leibniz Int. Proceedings in Informatics*, pages 253–269. Dagstuhl Publishing, Saarbrücken/Wadern, 2015.

[7] Maciej Piróg and Sam Staton. Backtracking with cut via a distributive law and left-zero monoids. *J. Funct. Program.*, 27, 2017.

[8] Gordon Plotkin and John Power. Algebraic operations and generic effects. *Appl. Categ. Struct.*, 11:69–94, 2003.

[9] A.L. Smirnov. Graded monads and rings of polynomials. *J. Math. Sci.*, 151(3):3032–3051, 2008.