A formal language for cyclic operads

Pierre-Louis Curien[1] and Jovana Obradović[2]

[1] πr^2 team, IRIF, CNRS, Université Paris Diderot, and Inria, France
curien@pps.univ-paris-diderot.fr
[2] πr^2 team, IRIF, CNRS, Université Paris Diderot, Inria, France, and University of Novi Sad, Serbia
jovana@pps.univ-paris-diderot.fr

An operad is a collection of abstract operations of different arities, equipped with a notion of how to compose them and an action of permuting their inputs. Operads encode categories of algebras whose operations have multiple inputs and one output, such as associative algebras, commutative algebras, Lie algebras, etc. The interest in encoding more general algebraic structures was a part of the *renaissance of operads* in the early nineties of the last century, when various generalizations of operads came into existence. The formalism of cyclic operads was introduced by Getzler and Kapranov in [2]. The motivation came from the framework of cyclic homology: in their paper, Getzler and Kapranov show that, in order to define a cyclic homology for $\mathcal{O}$-algebras, $\mathcal{O}$ has to be what they call a cyclic operad. The enrichment of the (ordinary) operad structure is provided by adding to the action of permuting the inputs of an operation an action of interchanging its output with one of the inputs, in a way that is compatible with operadic composition.

The notion of a cyclic operad was originally given in the *unbiased* manner in [2, Definition 2.1], over the structure of a monad in a category of unrooted trees. These trees act as pasting schemes, and the operations decorating their nodes are “composed in one shot” through the structure morphism of the algebra. Like operads, *biased* cyclic operads can be defined by means of simultaneous compositions [2, Theorem 2.2] or of partial composition [5, Proposition 42]. The fact that two operations can now be composed by grafting them along wires that “used to be outputs” leads to another point of view on cyclic operads, in which they are seen as generalisations of operads for which an operation, instead of having inputs and an (exchangeable) output, now has “entries”, and it can be composed with another operation along any of them. One can find such an *entries-only* definition in [4, Definition 48]. By contrast, we refer to the definitions based on describing cyclic operads as operads with extra structure as *exchangeable-output* ones.

The equivalence between the unbiased and biased definitions of a cyclic operad is formally given as a categorical equivalence that is, up to some extent, taken for granted in the literature. The issue that the construction of the structure morphism of an algebra over the monad out of the data of a biased cyclic operad should be shown independent of the way trees are decomposed has not been addressed in the proof of [2, Theorem 2.2], while the proof of [5, Proposition 42] is not given. Also, the monad structure is usually not spelled out in detail, in particular for what regards the correct treatment of the identities. The primary goal of this work is to formalise rigorously the equivalence between the unbiased and biased definitions of cyclic operads. Instead of comparing one of the two exchangeable-output biased definitions with the unbiased one, as done in [2, Theorem 2.2] and [5, Proposition 42], we show that the entries-only and the unbiased definition describe the same structure. Another particularity in our approach is that the appropriate categorical equivalence will be proved in a syntactical environment: a cyclic operad with biased composition will be expressed as a model of the equational theory determined by the axioms of the entries-only definition, while the monad of unrooted trees figuring in the unbiased approach will be expressed through a formal language called μ*-syntax*. Although μ-syntax was originally designed precisely to help us carry out this proof, it certainly has a value at the very level of encoding the (somewhat cumbersome) laws of the partial composition operation for cyclic operads. In other words, we also propose it as an alternative representation of the biased structure of a cyclic operad. We see this work as an experiment in bringing syntactical and type-theoretical know-how in the formal study of other algebraic structures used in connection with higher categories.

The name and the language of the μ-syntax formalism are motivated by another formal syntactical tool, the $\mu\tilde{\mu}$-subsystem of the $\overline{\lambda}\mu\tilde{\mu}$-calculus, presented by Curien and Herbelin in [1]. In their paper, programs are described by means of expressions called commands, of the form

$$\langle \mu\beta.c_1 \,|\, \tilde{\mu}x.c_2\rangle,$$

which exhibit a computation as the result of an interaction between a term $\mu\beta.c_1$ and an evaluation context $\tilde{\mu}x.c_2$, together with a symmetric reduction system

$$c_2[\mu\beta.c_1/x] \longleftarrow \langle \mu\beta.c_1 \,|\, \tilde{\mu}x.c_2\rangle \longrightarrow c_1[\tilde{\mu}x.c_2/\beta],$$

reflecting the duality between call-by-name and call-by-value evaluation. In our syntactical approach, we follow this idea and view operadic composition as such a program, i.e. as an interaction between two operations f and g, where f provides an input β (selected with μ) for the output x of g (marked with $\tilde{\mu}$). By moving this concept to the entries-only setting of cyclic operads, the input/output distinction of the $\mu\tilde{\mu}$-subsystem goes away, leading to the existence of a *single binding operator* μ, whose purpose is to select the entries of two operations which are to be connected in this interaction. More precisely, the expressions of the μ-syntax are

$$c ::= \langle s \,|\, t\rangle \;\mid\; \underline{f}\{t_{x_i} \,|\, i\in\{1,\dots,n\}\} \qquad s,t ::= x \;\mid\; \mu x.c$$

typed as follows

$$\frac{}{\{x\}\,|\,x} \qquad \frac{f\in\mathcal{C}(\{x_1,\dots,x_n\}) \quad Y_{x_i}\,|\,t_{x_i} \text{ for all } i\in\{1,\dots,n\}}{\underline{f}\{t_{x_i}\,|\,i\in\{1,\dots,n\}\} : \bigcup_{i=1}^n Y_{x_i}} \qquad \frac{X\,|\,s \quad Y\,|\,t}{\langle s\,|\,t\rangle : X\cup Y} \qquad \frac{c : X \quad x\in X}{X\backslash\{x\}\,|\,\mu x.c}$$

and the equations are

$$\langle s\,|\,t\rangle = \langle t\,|\,s\rangle \qquad \langle \mu x.c\,|\,s\rangle = c[s/x] \qquad \mu x.c = \mu y.c[y/x] \qquad \underline{f}\{t_x\,|\,x\in X\} = \underline{f^{\sigma}}\{t_{\sigma(y)}\,|\,y\in Y\}$$

The action of putting in line the characterization of the monad of unrooted trees, built upon the formalism of *unrooted trees with half-edges* commonly used in the operadic literature, together with the characterization by means of μ-syntax, makes the greatest part of the work. It involves setting up an intermediate formalism of unrooted trees, called the formalism of *Vernon trees*, that provides concise and lightweight pasting shemes for cyclic operads, and whose syntactical flavour reflects closely the shape of normal forms of the μ-syntax. The formal characterisation of a Vernon tree captures precisely the information relevant for describing the corresponding monad, which eases the verifications of the appropriate laws.

References

[1] P. -L. Curien, H. Herbelin, The duality of computation, ACM SIGPLAN Notices, Volume 35 Issue 9, 233-243, September 2000.

[2] E. Getzler, M. Kapranov, Cyclic operads and cyclic homology, Geom., Top., and Phys. for Raoul Bott, International Press, Cambridge, MA, 167-201, 1995.

[3] M. Markl, Models for operads, Comm. Algebra, 24(4):14711500, 1996.

[4] M. Markl, Modular envelopes, OSFT and nonsymmetric (non-Σ) modular operads, arXiv:1410.3414.

[5] M. Markl, Operads and PROPs, arXiv:math/0601129.

[6] J. P. May, The geometry of iterated loop spaces, volume 271 of Lectures Notes in Mathematics. Springer-Verlag, Berlin, 1972.