Operads, clones, and distributive laws

Pierre-Louis Curien

November 13, 2008

Abstract

We show how non-symmetric operads (or multicategories), symmetric operads, and clones, arise from three suitable monads on **Cat**, each extending to a monad on profunctors thanks to a distributive law. The presentation builds upon recent work by Fiore, Gambino, Hyland, and Winskel on a theory of generalized species of structures, but, for the multicategory case, the general idea goes back to Burroni's T-categories (1971). We show how other previous categorical analyses of operad (via Day's tensor products, or via analytical functor) fit with the profunctor approach.

1 Introduction

Operads, in their coloured and non-symmetric version, are also known as multicategories, since they are like categories, but with morphisms which have a single target as codomain but a multiple source (more precisely, a list of sources) as domain. These morphisms are often called operations (whence the term "operad" for the whole structure). Operads for short are the special case where the multicategory has just one object, and hence the arity of a morphism is just a number n (the length of the list), while the coarity is 1. There are many variations.

- non-symmetric operads versus symmetric operads versus clones (these variations concern the way in which operations are combined to form compound operations);
- operations versus cooperations (one input, several outputs), or "biop-erations" (several inputs, several outputs);
- operations whose shape, or arity, is more structured than a list (it could be a tree, etc...).

In this paper, we deal principally with the first variation, and touch on the second briefly. Our goal is to contribute to show that these variations can be smoothly and rather uniformly understood using the abstraction provided by monads.

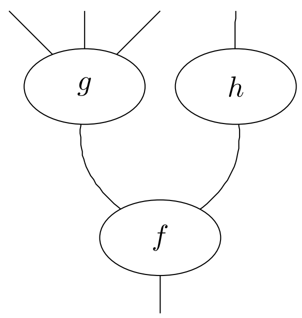
We are aware of two approaches for a general categorical account of such variations. The first one is based on spans, while the second one is based on profunctors. In both approaches, one abstracts the details of the variant in a monad, and one then extends the monad to the category of spans, or to the category of profunctors. Here, we take the profunctor approach. But the first monadic approach to multicategories, due to Burroni, is based on spans.

Spans are pairs of morphisms with the same domain, in a suitable category $\mathbf{C}$. Burroni has shown [7] that every cartesian monad T on $\mathbf{C}$ extends to a monad on spans, and then multicategories, for example, arise as endomorphisms endowed with a monoid structure in the (co)Kleisli (bi)category of a category of spans. A monad T is cartesian if it preserves pullbacks (hence $\mathbf{C}$ is required to have pullbacks) and if the naturality squares of the unit and multiplication of the monad are pullbacks.

Profunctors are functors $\Phi : \mathbf{C} \times \mathbf{C}'^{op} \to \mathbf{Set}$. As observed, say in [11], every monad in **Cat** (the category of categories) satisfying a certain distributive law can be extended to the category of profunctors (whose objects are categories and whose morphisms are profunctors). We show here how non symmetric and symmetric operads, and clones, arise (again) as monoids in the (co)Kleisli (bi)category associated with such an extended monad.

This is all rather "heavy" categorical vocabulary. We shall unroll this slowly in what follows. We offer the following intuitions for why monads, spans, and profunctors are relevant here.

- Monad. Consider the following composition of operations:



  By plugging the output of g and h on the inputs of f in parallel, we put together the three inputs of g and the input of h, yielding a compound operation with $3 + 1 = 4$ inputs. In other words, one may read two shapes on the upper part of the picture: $((\cdot,\cdot,\cdot),\cdot)$, and $(\cdot,\cdot,\cdot,\cdot)$. The first one remembers the construction, the second one flattens it. This flattening is a typical monad multiplication (monad of powersets, of lists, etc...).

- Span. Each bioperation has an arity and a coarity: the legs of the span are this arity and coarity mappings, respectively.

- Profunctor: A profunctor gives a family of sets of (pr)operations of fixed arity and coarity.

Arities (or coarities) are governed by the monad T. For example, an operad will be a span / profunctor from $T\mathbf{1}$ to $\mathbf{1}$, where T takes care of the multiplicity of inputs ($\mathbf{1}$ is the category with one object and one morphism). Similarly, a cooperad will go from 1 to $T\mathbf{1}$.

The span approach and the profunctor approach should be related, since one goes from profunctors to spans via the "element" or so-called Grothendieck construction. But under this correspondence the (bi)category of profunctors is (bi)equivalent to a subcategory of spans only, the discrete fibrations, while on the other hand the span approach leaves a lot of freedom on the choice of the underlying category $\mathbf{C}$. As a matter of fact, the two approaches have led to different types of successes. The current state of the art seems to be that:

1. using spans, an impressive variety of shapes in the *non-symmetric case* have been covered. We refer to [18] for a book-length account;

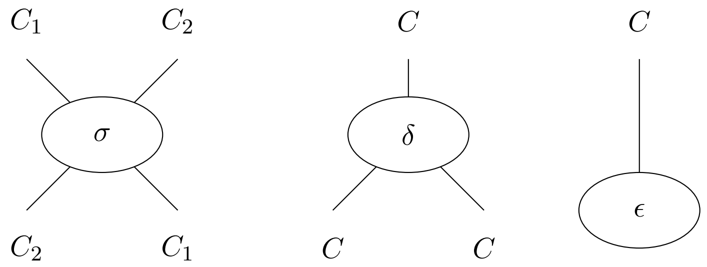
2. using profunctors one may cover the two other kinds of variations mentioned in this introduction.

In the sequel, we use (and introduce) the profunctor road.

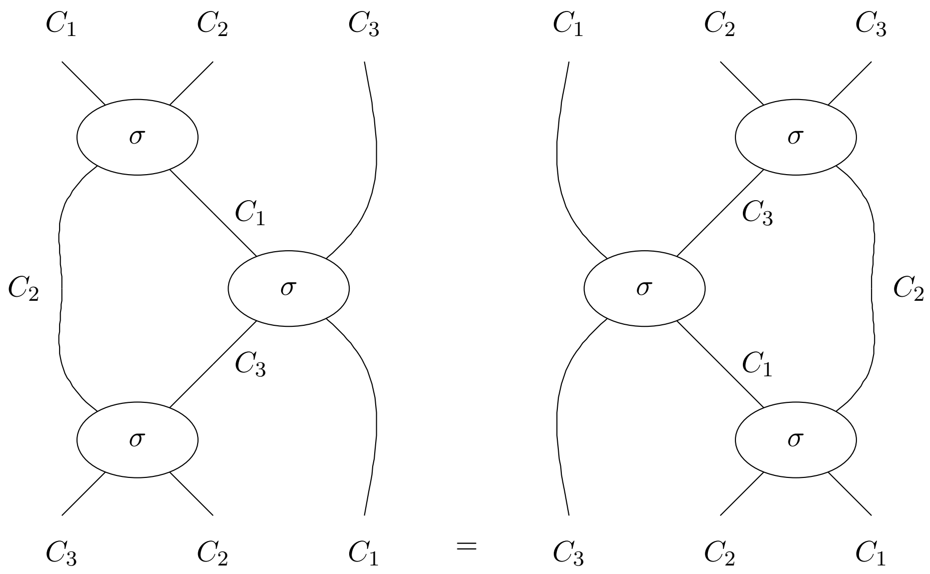
In Section 2, we introduce several monads on **Cat**. After recalling the notion of Kan extension in Section 3, we present two classical categorical accounts of operads in Sections 4 and 5. We proceed then to profunctors in Section 6. A plan of the rest of the paper is given at the end of that section.

2 Three useful combinators

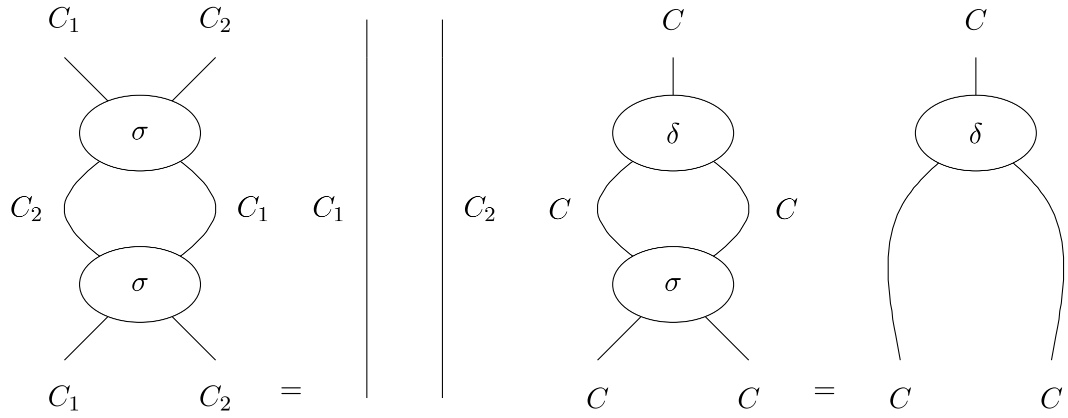
Consider the following three operations, or combinators. Think of them as morphisms in a category whose objects are (possibly empty) sequences $(C_1, \ldots, C_n)$, where the C_i's range over the objects of some category $\mathbf{C}$.



These combinators satisfy equations:

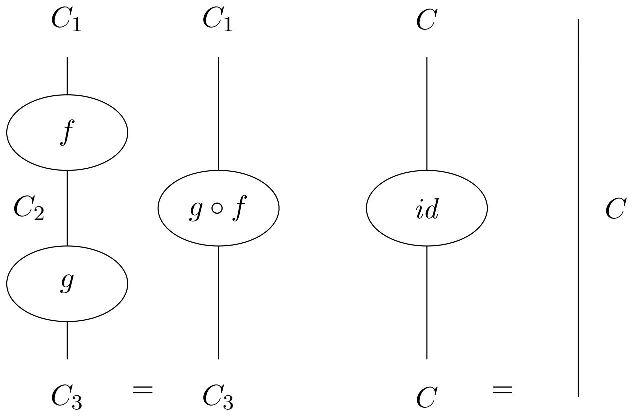


(compare with the familiar equations of transpositions, and notice that here we do not have to index them over natural numbers).



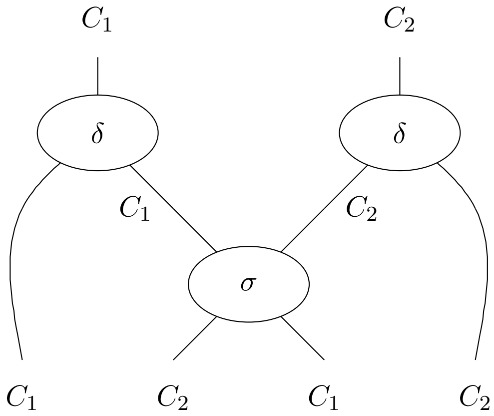
The other equations are the familiar ones for comonoids, and (mutatis mutandis) for distributive laws (see below). For a complete list, we refer the reader to [8, 17].

Take any combination X of these three combinators σ, δ, ϵ. We build the category $!_X\mathbf{C}$ as follows: objects are sequences $(C_1, \ldots, C_n)$ of objects of $\mathbf{C}$. Morphisms are *string diagrams* built out of the combinators taken from X and out of the morphisms of $\mathbf{C}$, quotiented over all the equalities that concern the combinators of X, including their naturality, and the equalities



which allow to embed $\mathbf{C}$ functorially into $!_X\mathbf{C}$. String diagrams are combinations of vertical and horizontal combinations of the basic combinators $\sigma, \delta, \epsilon, f$.

For six of the eight choices of X, the functor $!_X$ is equipped with a monad structure: the unit takes C to (C) and $f : C \to d$ to $f : (c) \to (d)$ and the multiplication on objects is the usual flattening, which takes $((C_1^1, \ldots, C_1^{i_1}), (C_n^1, \ldots, C_n^{i_n}))$ to $(C_1^1, \ldots, C_n^{i_n})$. A little care is needed to define the "flattening" on arrows. A morphism of $!!\mathbf{C}$ is an assembly of boxes connected by combinators typed in $!\mathbf{C}$. When we remove the boxes, we need to turn these combinators into (assemblings of) combinators typed in $\mathbf{C}$. We call this an expansion. A σ is expanded by means of σ's, an ϵ is expanded by putting ϵ's in parallel, but we need δ's *and* σ's to expand a δ. For example, at type (C_1, C_2), the expansion of $\delta : ((C_1, C_2)) \to ((C_1, C_2), (C_1, C_2))$ is the following morphism from (C_1, C_2) to (C_1, C_2, C_1, C_2):



We therefore exclude $!_{\{\delta\}}$ and $!_{\{\delta,\epsilon\}}$ from our treatment. But all the other ! constructions are fine.

Three of the six remaining combinations are of particular interest. If $X = \emptyset$ (resp. $X = \{\sigma\}$, $X = \{\sigma, \delta, \epsilon\}$), then $!\mathbf{C}$ is the free monoidal (resp. symmetric monoidal, cartesian) category over $\mathbf{C}$. (By cartesian category, we mean "category with finite products"). We write them $!_m, !_s, !_f$, respectively:

monoidal	symmetric monoidal	cartesian
$!_m$	$!_s$	$!_f$

When $X = \{\sigma, \delta, \epsilon\}$, the !-algebras are the cartesian categories, and when $X = \emptyset$ (resp. $X = \{\sigma\}$), the (pseudo-) !-algebras are the monoidal (resp. symmetric monoidal) categories. We refer to [18] for details.

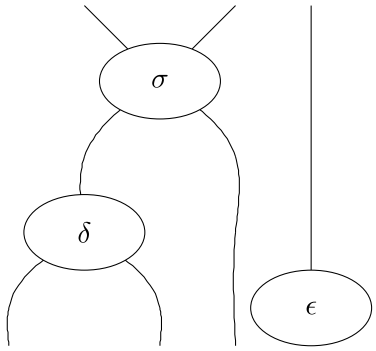
We define $?\mathbf{C} = (!(\mathbf{C}^{op}))^{op}$. Graphically, this amounts to reversing the basic combinators σ, δ, ϵ, which could then, if we cared, be called σ, μ, η while maintaining the orientation of the combinators f imported from $\mathbf{C}$ (since they become themselves again after two ^{op}'s. But it is more convenient to stick with σ, δ, ϵ, to reverse the direction of the f's, and to read the diagrams in the bottom-to-top direction. In particular, when $\mathbf{C} = \mathbf{1}$, there is no f to reverse...

When $X = \emptyset$ or $X = \{\sigma\}$, then $?\mathbf{C}$ is (isomorphic to) $!\mathbf{C}$. When $X = \{\sigma, \delta, \epsilon\}$, $?\mathbf{C}$ is the free cocartesian category over $\mathbf{C}$. The objects of $?\mathbf{1}$ are written $0, 1, 2, \ldots$, standing for $(), (\cdot), (\cdot, \cdot, \cdot), \ldots$, where $\cdot$ is the unique object of $\mathbf{1}$.

Note that for any choice of ?, there is a faithful functor from $?\mathbf{1}$ to $\mathbf{Set}$. This function is obtained as follows. For any morphism from m to n, i.e., for any string diagram constructed out of the combinators in X that has n input wires and m output wires, one constructs its interpretation as a function from $\{0, \ldots, m-1\}$ to $\{0, \ldots, n-1\}$, by the following rules.

- the function associated with σ is the transposition on $\{0, 1\}$;
- the function associated with ϵ is the unique function from the empty set to $\{0\}$;
- the function associated with δ is the unique function from $\{0, 1\}$ to $\{0\}$;
- the function associated with a vertical composition of two string diagrams is their composition;
- the function associated with a horizontal composition of two string diagrams is their categorical sum, i.e., if the two diagrams define $f : \{0, \ldots, m-1\} \to \{0, \ldots, n-1\}$ and $g : \{0, \ldots, p-1\} \to \{0, \ldots, q-1\}$, then the whole diagram defines $f+g : \{0, \ldots, m+p-1\} \to \{0, \ldots, n+q-1\}$, where $(f+g)(i) = f(i)$ if $i < m$ and $f + g(i) = g(i) + n$ if $i \geq m$.

More synthetically, the function associated with a string diagram is obtained by naming the output and input wires $0, 1, \ldots, m_1$ and $0, 1, \ldots, n-1$, respectively, and then computing $f(i)$ as the name of the input wire reached when starting from input wire i, going up in δ nodes, and going up left or right in σ nodes according to whether this node is reached from down right or down left, respectively. For example, the picture



represents the function $f : \{0, 12\} \to \{0, 1, 2\}$ defined by

$$f(0) = f(1) = 1 \quad f(2) = 0$$

Note on this example that ϵ witnesses the lack of surjectivity. More precisely, as first shown by Burroni, the functor, when suitably corestricted, gives the following isomorphims and equivalences of categories:

$?_{\emptyset}$	identity functions
$?_{\{\sigma\}}$	bijections
$?_{\{\sigma,\delta,\epsilon\}}$	all functions
$?_{\{\delta,\epsilon\}}$	monotone functions
$?_{\{\sigma,\delta\}}$	surjective functions
$?_{\{\sigma,\epsilon\}}$	injective functions

where, say, the last line should be read as follows: the category $?_{\{\sigma,\epsilon\}}(\mathbf{1})$ is equivalent to the category of finite sets and injective functions, and isomorphic to its full subcategory whose objects are the sets $\{0, \ldots, i-1\}$ ($i \in \mathbb{N}$). For $?_{\{\delta,\epsilon\}}(\mathbf{1})$, one must take the category of all finite total orders.

3 Kan extensions

We recall that given a functor $K : \mathbf{M} \to \mathbf{C}$, a left Kan extension of a functor $T : \mathbf{M} \to \mathbf{A}$ along K is a pair of a functor $Lan_K(T) : \mathbf{C} \to \mathbf{A}$ and a natural transformation $\eta_T : T \to Lan_K(T)K$ such that $(Lan_K(T), \eta_T)$ is universal from T to $\mathbf{A}^K$. This means that for any other pair of a functor $S : \mathbf{C} \to \mathbf{C}$ and a natural transformation $\alpha : T \to SK$ there exists a unique natural transformation $\mu : Lan_K(T) \to S$ such that $\alpha = \mu K \circ \eta$. Here we shall only need to know that the solution of this universal problem is given by the following formula for $Lan_K(T)$:

$$Lan_K Tc = \int^m \mathbf{C}[Km, c] \cdot Tm$$

where we use Mac Lane's notation for coends [19]. Coends are sorts of colimits (or inductive limits), adapted to the case of diagrams which vary both covariantly and contravariantly over some parameter: here, m appears contravariantly in $\mathbf{C}[Km, c]$ and covariantly in Tm. In the formula, $\mathbf{C}[Km, c] \cdot Tm$ stands for the coproduct of as many copies of Tm as there are morphisms from Km to c.

$$\begin{tikzcd}
\mathbf{M} \arrow[r, "K"] \arrow[d, "T"'] & \mathbf{C} \arrow[dl, "Lan_K T"] \\
\mathbf{A} &
\end{tikzcd}$$

When K is full and faithful, then η is iso, and in particular the triangle commutes (up to isomorphism). When moreover K is the Yoneda embedding, we get

$$Lan_{\mathcal{Y}} TX = \int^m Xm \cdot Tm$$

When moreover $\mathbf{A}$ is a presheaf category $\mathbf{Set}^{\mathbf{D}}$, we have (since limits and colimits of presheaves are pointwise):

$$Lan_{\mathcal{Y}} TXd = \int^m Xm \times Tmd$$

or

$$Lan_{\mathcal{Y}} TXd = (\Sigma_m Xm \times Tmd)/ \approx$$

When $\approx$ is termwise, i.e. is the disjoint union of equivalences $\approx_m$ each on $Xm \times Tmd$, we get a formula which will look more familiar:

$$Lan_{\mathcal{Y}} TXd = \Sigma_m (Xm \otimes_m Tmd)$$

(where we have writen $(Xm \times Tmd)/{\approx_m}$ as $Xm \otimes_m Tmd$). In our examples, $\approx$ will not always be termwise (it will be in the operad cases, but not in the clone case).

4 Kelly's account of operads

In 1972, Kelly gave the following description of operads [16] (see also [10]):

$$\begin{tikzcd}
\mathbf{1} \arrow[r] \arrow[dr, "X"'] & !\mathbf{1} \arrow[r, "\mathcal{Y}"] \arrow[d, "X^{\otimes\text{-}}"] & \mathbf{Set}^{?\mathbf{1}} \arrow[dl, "Lan_{\mathcal{Y}}(X^{\otimes\text{-}}) = _\bullet X"] \\
& \mathbf{Set}^{?\mathbf{1}} &
\end{tikzcd}$$

In this diagram, the functor $X^{\otimes\text{-}}$ associates with n the nth iterated tensor of X, with respect to the following monoidal structure on $\mathbf{Set}^{?\mathbf{1}}$, due (in a more general setting) to Day [9]:

$$(X \otimes Y)p = \int^{m,n} Xm \times Yn \times ?\mathbf{1}[m+n,p] \qquad I_{Day}n = ?\mathbf{1}[0,n]$$

The n-ary tensor product is described by the following formula:

$$(X_1 \otimes X_2 \otimes \ldots \otimes X_m)p = \int^{n_1,\ldots,n_m} X_1 n_1 \times \ldots \times X_m n_m \times ?\mathbf{1}[n_1 + \ldots + n_m, p]$$

Hence the formula for $Lan_{\mathcal{Y}}(X^{\otimes}\text{-})Y$, which we write $Y \bullet X$, is

$$\begin{array}{rcl} (Y \bullet X)p & = & \int^m Ym \times X^{\otimes m}p \\ & = & \int^{m,n_1,\ldots,n_m} Ym \times Xn_1 \times \ldots \times Xn_m \times ?\mathbf{1}[n_1 + \ldots + n_m, p] \end{array}$$

In the case of $!_f$ this boils down to

$$(Y \bullet X)p = \int^m Ym \times (Xp)^m$$

This is because in a cocartesian category, an arrow $f : n_1 + \ldots + n_m \to p$ amounts to arrows $f_i \in ?\mathbf{1}[n_i, p]$ which allows us to define a map from $Xn_1 \times \ldots \times Xn_m \times ?\mathbf{1}[n_1 + \ldots + n_m, p]$ to $(Xp)^m$ as follows:

$$(a_1, \ldots, a_m, f) \mapsto (Xf_1 a_1, \ldots, Xf_m a_m)$$

and then to "extend" cocones indexed over m to cocones indexed over $m, n_1, \ldots, n_m$.

One can check rather easily that the operation $\bullet$, together with the unit defined by

$$In = ?\mathbf{1}[1,n]$$

provide a (non-symmetric) monoidal structure on $\mathbf{Set}^{?\mathbf{1}}$ (for $I \cdot X = X$, one uses the fact that $X^{\otimes}\text{-}$ is functorial).

When ! is $!_m, !_s, !_f$, respectively, a monoid for the structure $(I, \bullet)$ is an operad, a symmetric operad, a clone, respectively.

5 Operads from analytic functors

Another way to define the composition operation $\bullet$ is via a different Kan extension:

$$\begin{tikzcd} ?\mathbf{1} \arrow[r, "\subseteq"] \arrow[d, "X"'] & \mathbf{Set} \arrow[dl, "Lan_{\subseteq} X"] \\ \mathbf{Set} & \end{tikzcd}$$

where $\subseteq$ is the faithful (and non full) functor described at the end of Section 2. This is the approach taken by Joyal (for $!_s$) [15]. In Joyal's language, X is called a *species of structure*, and $Lan_{\subseteq} X$ is the associated *analytic functor*, whose explicit formula is (for any set z):

$$Lan_{\subseteq} X z = \int^m z^m \times Xm$$

Joyal's approach requires more apparatus, namely the fact that $Lan_{\subseteq}$ is faithful. $Y \bullet X$ is characterized by the following equality:

$$Lan_{\subseteq}(Y \bullet X) = Lan_{\subseteq} Y \circ Lan_{\subseteq} X$$

and we have indeed:

$$\begin{array}{l} Lan_{\subseteq} Y(Lan_{\subseteq} X z) \\ \quad = \int^m (\int^n z^n \times Xn)^m \times Ym \\ \quad = \int^{m,n_1,\ldots,n_m} z^{n_1+\ldots+n_m} \times Xn_1 \times \ldots \times Xn_m \times Ym \\ \quad = \int^p z^p \times (\int^{m,n_1,\ldots,n_m} Ym \times Xn_1 \times \ldots \times Xn_m \times ?\mathbf{1}[n_1 + \ldots + n_m, p]) \\ \quad = Lan_{\subseteq}(Y \bullet X) z \end{array}$$

(the summation over p is superfluous since from $f \in ?\mathbf{1}[n_1 + \ldots + n_m, p]$ we get $z^{\subseteq(f)}$ from z^p to $z^{n_1+\ldots+n_m}$).

6 Profunctors

Recall that a profunctor (or distributor) [4], see also [5]) Φ from $\mathbf{C}$ to $\mathbf{C}'$ is a functor

$$\Phi : \mathbf{C} \times \mathbf{C}'^{op} \to \mathbf{Set}$$

We write $\Phi : \mathbf{C} \rightarrow\!\!\!\!\!+\, \mathbf{C}'$. Composition of profunctors is given by the following formula:

$$(\Psi \circ \Phi)(C, C'') = \int^{C'} \Psi(C', C'') \times \Phi(C, C')$$

Therefore, profunctors compose only up to isomorphism. Categories, profunctors, and natural transformations form thus, not a 2-category, but a bicategory [3].

The (pseudo-)category $\mathbf{Prof}$ of profunctors is self-dual, via the isomorphism ${}^{op} : \mathbf{Prof} \to \mathbf{Prof}^{op}$ which maps $\mathbf{C}$ to $\mathbf{C}^{op}$ and $\Phi : \mathbf{C_1} \rightarrow\!\!\!\!\!+\, \mathbf{C_2}$ to

$$\Phi^{op} = ((C_2, C_1) \mapsto \Phi(C_1, C_2)) : \mathbf{C_2}^{op} \rightarrow\!\!\!\!\!+\, \mathbf{C_1}^{op}$$

The composition of profunctors can be synthesized via Kan extensions: Given $\Phi : \mathbf{C} \rightarrow\!\!\!\!\!+\, \mathbf{C}'$ and $\Psi : \mathbf{C}' \rightarrow\!\!\!\!\!+\, \mathbf{C}''$, consider their "twisted curried"[1] versions $\Phi' : \mathbf{C}'^{op} \to \mathbf{Set}^{\mathbf{C}}$ and $\Psi' : \mathbf{C}''^{op} \to \mathbf{Set}^{\mathbf{C}'}$, defined by $\Phi' C' C =$

[1]After the name of Curry, who defined a calculus of functions called combinatory logic, based on application and a few combinators, and where functions of several arguments are expressed through repeated applications.

$\Phi(C, C')$ and $\Psi' C'' C' = \Psi(C', C'')$. Then it is immediate to check that $\Psi \circ \Phi$ is the uncurried version of $(Lan_{\mathcal{Y}}\Phi') \circ \Psi'$, as illustrated in the following diagram:

$$\begin{tikzcd}
\mathbf{C}'^{op} \arrow[r, "\mathcal{Y}"] \arrow[d, "\Phi'"] & \mathbf{Set}^{\mathbf{C}'} \arrow[dl, "Lan_{\mathcal{Y}}\Phi'"] & \mathbf{C}''^{op} \arrow[l, "\Psi'"'] \arrow[dll, "(\Psi\circ\Phi)'"] \\
\mathbf{Set}^{\mathbf{C}} & &
\end{tikzcd}$$

Now we take "profunctor glasses" to look at Kelly's and Joyal's diagrams.

- A presheaf $X : ?\mathbf{1} \to \mathbf{Set}$ can be viewed as a profunctor

$$(n \mapsto (\cdot \mapsto Xn)) : ?\mathbf{1} \nrightarrow \mathbf{1}$$

- $X^{\otimes\text{-}}$ can be viewed as a profunctor from $?\mathbf{1}$ to $?\mathbf{1}$ (since $(?\mathbf{1})^{op} = !(\mathbf{1}^{op}) = !\mathbf{1}$). Thus the transformation

$$\frac{X : ?\mathbf{1} \nrightarrow \mathbf{1}}{X^{\otimes\text{-}} : ?\mathbf{1} \nrightarrow ?\mathbf{1}}$$

  suggests to consider ? as a comonad over **Prof**.

- The formula defining $Y \bullet X$ describes the profunctor composition of $Y : ?\mathbf{1} \nrightarrow \mathbf{1}$ and $X^{\otimes\text{-}} : ?\mathbf{1} \nrightarrow ?\mathbf{1}$ (cf. the Kan extension explanation of the composition of profunctors given above). But a better way to put it is that $Y \bullet X$ is the compostion of Y and X in the coKleisli category $\mathbf{Prof}_?$, and we shall see that it is indeed the case.

- Joyal's construction amounts to "taking points". We first observe that $?\emptyset = \mathbf{1}$, where $\emptyset$ is the empty category. (This holds for any of the eight choices for !.) It follows that $\emptyset$ is terminal in $\mathbf{Prof}_?$. In this category, the points of an object $\mathbf{C}$ are the morphisms from $\emptyset$ to $\mathbf{C}$, i.e., the profunctors from $\mathbf{1}$ to $\mathbf{C}$, i.e. the presheaves over $\mathbf{C}^{op}$. In particular, the points of $\mathbf{1}$ are just sets. We shall see that

$$\mathbf{Prof}_?[\emptyset, X] = Lan_{\subseteq} X$$

  i.e., that the analytic functor associated with the presheaf X describes its pointwise behaviour in $\mathbf{Prof}_?$.

Our goal is thus to figure out ? as a comonad on **Prof**. We shall do this in three steps:

- In Section 7, we show that profunctors arise as a Kleisli category, using the (pseudo-)monad Psh of presheaves.

- In Section 9, we show that all the monads ! of Section 2 distribute over *Psh*. Distributive laws are recalled in Section 8. In Section 10, we pause to explain how both Day's tensor product and this distributive law can be synthesized out of considerations of structure preservation.

- in Section 11, we show that this distributive law allows us to extend ! to a monad on **Prof**, and by self-duality we obtain the comonad ? that we are looking for.

Warning. In what follows, for simplicity, we ignore coherence and size issues. Coherence issues arise from the fact that we shall compose arrows using coend formulas, which make sense only up to iso. The qualifiers "(pseudo-)" or "(bi)" will be the only traces of this concern here. Size issues arise from the fact that we shall consider the presheaf construction as a monad on **Cat**. We refer to [20, 12] for indications on how these issues can be handled rigorously.

7 Profunctors as a Kleisli category

As we have already done when defining composition, we can alternatively write a profunctor $\Phi : \mathbf{C_1} \rightarrow\!\!\!\!+ \mathbf{C_2}$ as

$$C_1 \mapsto (C_2 \mapsto \Phi(C_1, C_2)) : \mathbf{C_1} \to \mathbf{Set}^{\mathbf{C_2}^{op}}$$

This suggests to look at the endofunctor *Psh* on **Cat** defined by $Psh(\mathbf{C}) = \mathbf{Set}^{\mathbf{C}^{op}}$. We show that *Psh* is a (pseudo-)monad, and that the above composition is the composition of the Kleisli category, so that profunctors arise as a Kleisli category. We do not know when this was first noticed. The action of *Psh* on functors is defined as follows:

$$\begin{tikzcd} \mathbf{C} \arrow[r, "\mathcal{Y}"] \arrow[d, "\mathcal{Y}\circ F=(C\mapsto(D\mapsto\mathbf{D}[D,FC]))"'] & \mathbf{Set}^{\mathbf{C}^{op}} \arrow[dl, "PshF=Lan_{\mathcal{Y}}(\mathcal{Y}\circ F)"] \\ \mathbf{Set}^{\mathbf{D}^{op}} & \end{tikzcd}$$

and the explicit formula is (for $F : \mathbf{C} \to \mathbf{D}$):

$$Psh(F)XD = \int^{C} XC \times \mathbf{D}[D, FC]$$

The unit η is $\mathcal{Y} : \mathbf{C} \to Psh(\mathbf{C})$ and the multiplication is defined (again) by a left Kan extension:

$$\begin{tikzcd} \mathbf{Set}^{\mathbf{C}^{op}} \arrow[r, "\mathcal{Y}"] \arrow[d, "id"'] & \mathbf{Set}^{(\mathbf{Set}^{\mathbf{C}^{op}})^{op}} \arrow[dl, "\mu=Lan_{\mathcal{Y}}(id)"] \\ \mathbf{Set}^{\mathbf{C}^{op}} & \end{tikzcd}$$

with explicit formula

$$\mu(H)C = \int^F HF \times FC$$

Let us check that the monad laws are satisfied. One of the neutrality rules is the commuting diagram defining μ. Here is the calculation for the other one:

$$\begin{array}{l}(\mu \circ Psh(\eta))(X) \\ \quad = \mu(Y \mapsto (C \mapsto \int^C XC \times \mathbf{Set}^{\mathbf{C}^{op}}[Y, \mathcal{Y}C])) \\ \quad = (C' \mapsto \int^Y \int^C XC \times \mathbf{Set}^{\mathbf{C}^{op}}[Y, \mathcal{Y}C] \times YC') \\ \quad = (C' \mapsto XC')\end{array}$$

where the (Y, C) component of the wedge (wedges are to (co)cones what (co)ends are to (co)limits) to XC' is $((u, \alpha, v) \mapsto X(\alpha C' v)u)$.
The two sides of the associativity equation for μ are (for Ξ, C fixed):

$$\begin{array}{l}\int^X \int^H \Xi H \times HX \times XC \\ \int^X \int^H \Xi H \times \mathbf{Set}^{\mathbf{C}^{op}}[X, \mu H] \times XC\end{array}$$

so that we are left to show (for H, C fixed):

$$\int^X \mathbf{Set}^{\mathbf{C}^{op}}[X, \mu H] \times XC = \int^X HX \times XC$$

One builds a wedge to the right hand side by taking $((\alpha, u) \mapsto \alpha C u)$ as X component. In the other direction, we use the map from HX to $\mathbf{Set}^{\mathbf{C}^{op}}[X, \mu H]$ which takes u to the natural transformation α defined by $\alpha C' v = [u, v]$ (where $v \in XC'$ and $[u, v]$ is the equivalence class of (u, v) in the quotient $(\Sigma^X HX \times XC')/\approx)$).

Finally, we check that the Kleisli composition gives back the composition of profunctors as given above (the definition of the composition in a Kleisli category is recalled in the next section). We have to compute

$$\mu \circ Psh(C' \mapsto (C'' \mapsto (\Psi(C', C'')))) \circ (C \mapsto (C' \mapsto (\Phi(C, C'))))$$

which is

$$\begin{array}{l}(C \mapsto \mu(X \mapsto \int^{C'} \Phi(C, C') \times \mathbf{Set}^{\mathbf{C}''^{op}}[X, (C'' \mapsto \Psi(C', C''))])) \\ \quad = (C \mapsto (C'' \mapsto \int^X \int^{C'} \Phi(C, C') \times \mathbf{Set}^{\mathbf{C}''^{op}}[X, (C'' \mapsto \Psi(C', C''))] \times XC'')) \\ \quad = (C \mapsto (C'' \mapsto \int^{C'} \Phi(C, C') \times (\int^X \mathbf{Set}^{\mathbf{C}''^{op}}[X, (C'' \mapsto \Psi(C', C''))] \times XC''))) \\ \quad = (C \mapsto (C'' \mapsto \int^{C'} \Phi(C, C') \times \Psi(C', C'')))\end{array}$$

The last step is justified by taking $Y = (C'' \mapsto \Psi(C', C''))$ in the following formula, which holds for any presheaf Y:

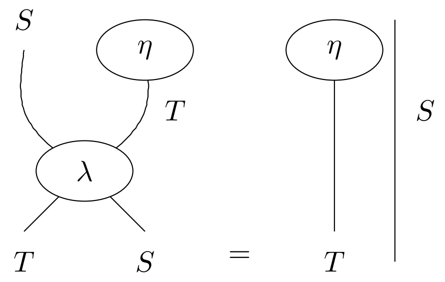
$$\int^X \mathbf{Set}^{\mathbf{C}''^{op}}[X, Y] \times XC'' = YC''$$

The component X of the coend wedge is $((\alpha, u) \mapsto \alpha C''u)$, and for any wedge from the diagram to some set z, we get the map from YC'' to z by considering its component at $\mathbf{C}''[\_, C'']$.

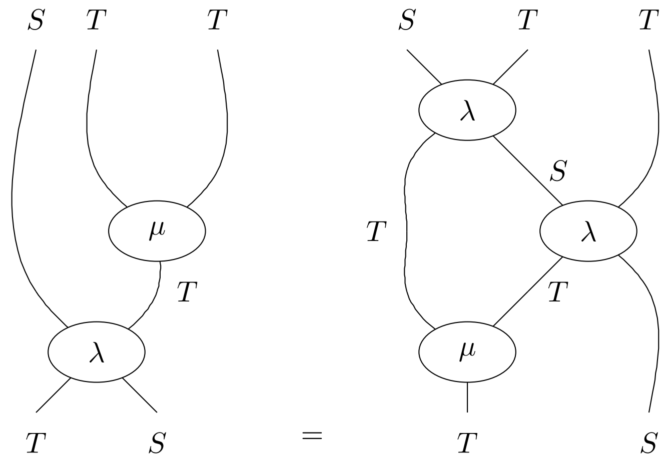
8 Distributive laws

Recall that a distributive law [2] (see also [1]) is a natural transformation $\boldsymbol{\lambda} : TS \to ST$, where (S, η_S, μ_S) and (T, η_T, μ_T) are two monads over the same category $\mathbf{C}$, satisfying the following laws (expressed using self-explanatory pictures of the same nature as the other ones used here):

EQUATION $(\lambda - \eta_T)$:



EQUATION $(\lambda - \mu_T)$:

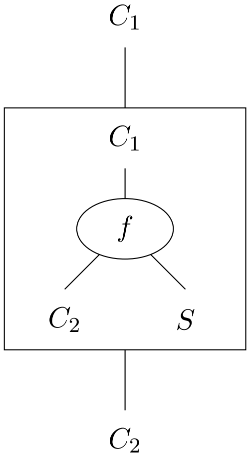


and two similar equations $(\lambda - \eta_S)$ and $(\lambda - \mu_S)$. A distributive law allows us to extend T to the Kleisli category of S (and conversely, such a lifting induces a distributive law such that the two constructions are inverse to each other).

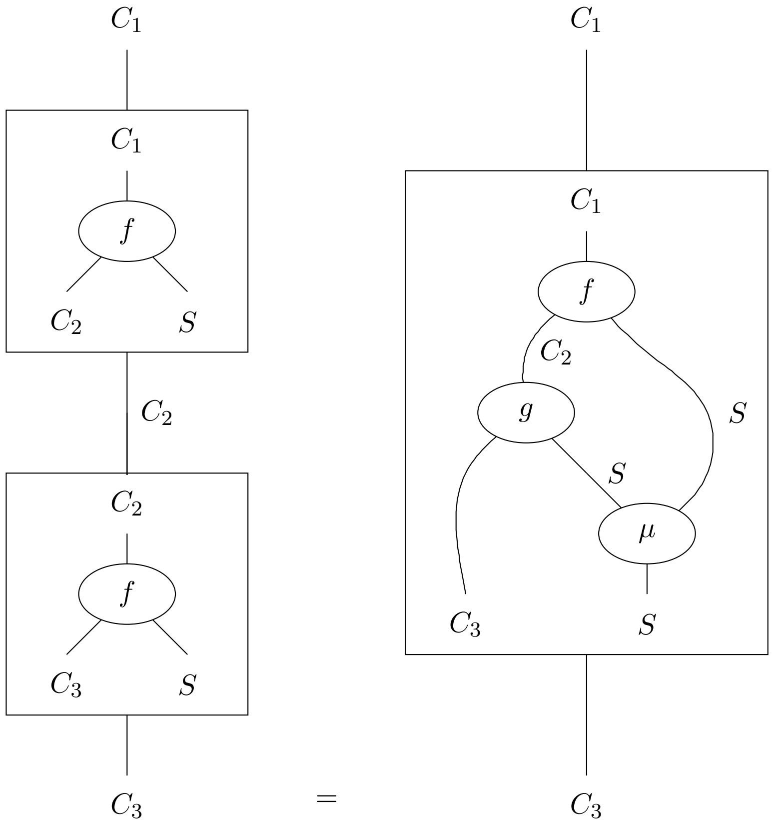
Let us first recall that the Kleisli category, notation $\mathbf{C}_S$, has the same objects as $\mathbf{C}$, while the morphisms are defined as follows:

$$\mathbf{C}_S[C_1, C_2] = \mathbf{C}[C_1, SC_2]$$

The identity morphism is given by η_S. We describe the composition pictorially. For this we represent a morphism in the Kleisli category as follows:

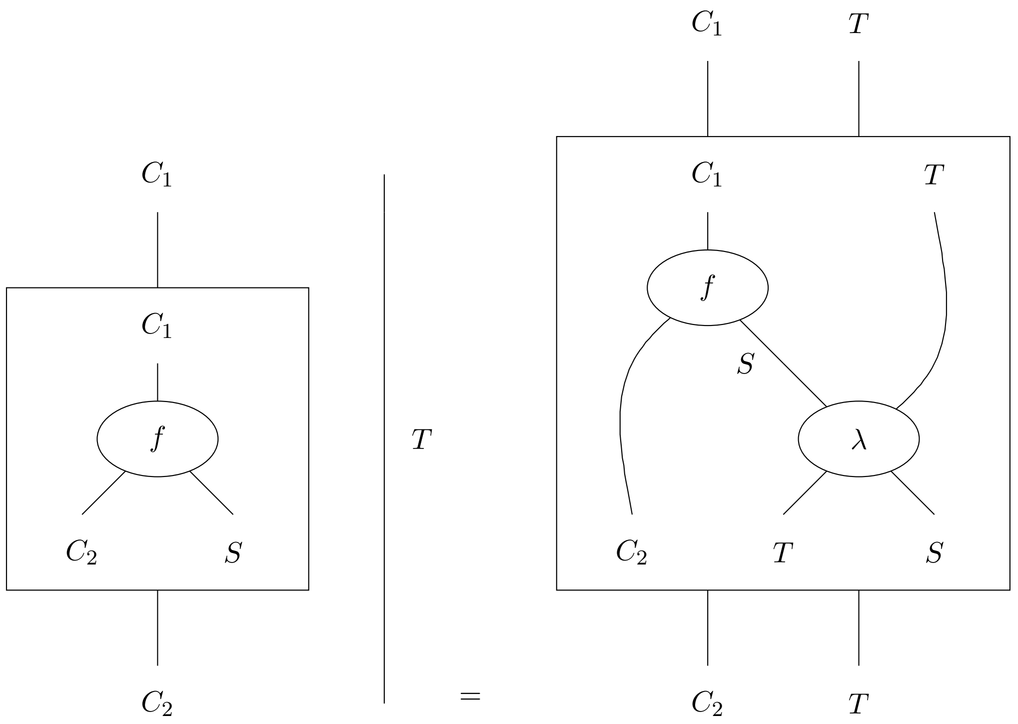


where the box separates the inside which lives in $\mathbf{C}$ from the outside which lives in $\mathbf{C}_S$. Composition is then defined as follows:



That this yields a category is equivalent to the requirement that S is a monad.

Now we can "extend" the endofunctor T to an endofunctor (still called T) on $\mathbf{C}_S$, as follows. The new T acts on objects as the old T. On morphisms, its action is described as follows:



It is a nice exercise to check that the old and the new T satisfy

$$T \circ F_S = F_S \circ T$$

where $F_S : \mathbf{C} \to \mathbf{C}_S$ is defined by $F_S(C) = C$ and $F_S(f) = \eta_S \circ f$, and that the new T is a monad.

9 A $!/Psh$ distributive law

We define a transformation $\boldsymbol{\lambda} : ! \circ Psh \to Psh \circ !$, as follows:

$$(F_1, \ldots, F_n) \mapsto ((C_1, \ldots, C_m) \mapsto \int^{D_1, \ldots, D_n} F_1 D_1 \times \ldots \times F_n D_n \times !\mathbf{C}[(C_1, \ldots, C_m), (D_1, \ldots, D_n)])$$

This transformation is functorial:

- $\alpha : F \to G$. There is an obvious map from $\int^D FD \times !\mathbf{C}[C_1, \ldots, C_m, (D)]$ to $\int^D GD \times !\mathbf{C}[C_1, \ldots, C_m, (D)]$.
- Combinators, say $\delta : (F) \to (F, F)$. We obtain a map

  from $\int^D FD \times !\mathbf{C}[C_1, \ldots, C_m, (D)]$
  to $\int^{D_1, D_2} FD_1 \times FD_2 \times !\mathbf{C}[(C_1, \ldots, C_m, (D_1, D_2)]$

by going

$$\text{from } FD\times !\mathbf{C}[C_1,\ldots,C_m,(D)]$$
$$\text{to } FD\times FD\times !\mathbf{C}[(C_1,\ldots,C_m,(D,D)]$$

We verify the four equations:

$(\boldsymbol{\lambda}-\eta_!)$. The left hand side is computed by taking the case $n=1$ in the definition of $\boldsymbol{\lambda}$, thus we obtain

$$F\mapsto ((C_1,\ldots,C_m)\mapsto \int^D FD\times !\mathbf{C}[(C_1,\ldots,C_m),(D)])$$

which is the formula for $Psh(\eta_!)$.

$(\boldsymbol{\lambda}-\eta_{Psh})$. We have (at $(A_1,\ldots,A_n)$, $(C_1,\ldots C_m)$):

$$\begin{array}{l}\int^{D_1,\ldots,D_n}\mathbf{C}[D_1,A_1]\times\ldots\times\mathbf{C}[D_n,A_n]\times !\mathbf{C}[(C_1,\ldots,C_m),(D_1,\ldots,D_n)]\\ \quad = !\mathbf{C}[(C_1,\ldots,C_m),(A_1,\ldots,A_n)]=\mathcal{Y}(A_1,\ldots,A_n)(C_1,\ldots C_m)\end{array}$$

$(\boldsymbol{\lambda}-\mu_!)$. The left hand side is (at $((F_1^1,\ldots,F_1^{i_1}),\ldots(F_n^1,\ldots,F_n^{i_n}))$, $(C_1,\ldots,C_m)$):

$$\int^{D_1^1,\ldots,D_n^{i_n}} F_1^1D_1^1\times\ldots\times F_n^{i_n}D_n^{i_n}\times !\mathbf{C}[\vec{C},(D_1^1,\ldots,D_n^{i_n})]$$

We first compute the upper part $(\boldsymbol{\lambda}!)\circ(!\boldsymbol{\lambda})$ of the right hand side (at $(\vec{F_1},\ldots,\vec{F_n})$, $((A_1^1,\ldots,A_1^{j_1}),\ldots,(A_p^1,\ldots,A_p^{j_p})))$:

$$\begin{array}{l}\int^{\vec{B_1},\ldots\vec{B_n}}\int^{D_1^1,\ldots,D_n^{i_n}} F_1^1D_1^1\times\ldots\times F_n^{i_n}D_n^{i_n}\times !\mathbf{C}[\vec{B_1},\vec{D_1}]\times\ldots\times !\mathbf{C}[\vec{C_n},\vec{D_n}]\times !!\mathbf{C}[\vec{\vec{A}},\vec{\vec{B}}]\\ \quad =\int^{D_1^1,\ldots,D_n^{i_n}} F_1^1D_1^1\times\ldots\times F_n^{i_n}D_n^{i_n}\times !!\mathbf{C}[\vec{\vec{A}},\vec{\vec{D}}]\end{array}$$

Finally, applying $Psh(\mu)$, we get (at $\vec{\vec{F}}$, $\vec{C}$):

$$\int^{\vec{A_1},\ldots,\vec{A_p}}\int^{D_1^1,\ldots,D_n^{i_n}} F_1^1D_1^1\times\ldots\times F_n^{i_n}D_n^{i_n}\times !!\mathbf{C}[\vec{\vec{A}},\vec{\vec{D}}]\times !\mathbf{C}[\vec{C},(A_1^1,\ldots,A_p^{j_p})]$$

and we conclude using $\mu_!$.

$(\boldsymbol{\lambda}-\mu_{Psh})$. The left hand side is (at $((H_1,\ldots,H_n)$, $(C_1,\ldots,C_m)$):

$$\begin{array}{l}\int^{D_1,\ldots,D_n}\int^{G_1,\ldots,G_n} H_1G_1\times G_1D_1\times\ldots\times H_nG_n\times G_nD_n\times !\mathbf{C}[\vec{C},\vec{D}]\\ \quad =\int^{G_1,\ldots,G_n} H_1G_1\times\ldots\times H_nG_n\times\boldsymbol{\lambda}\vec{G}\end{array}$$

The right hand side is

$$\int^K\int^{F_1,\ldots,F_p}\int^{G_1,\ldots,G_n} H_1G_1\times\ldots\times H_nG_n\times !\mathbf{Set}^{\mathbf{C}^{op}}[\vec{F},\vec{G}]\times\mathbf{Set}^{!\mathbf{C}^{op}}[K,\boldsymbol{\lambda}\vec{F}]\times K\vec{C}$$

The latter coend reduces to the former: in one direction, consider the maps

$$(\beta,\alpha,x)\mapsto(\boldsymbol{\lambda}\beta)(\alpha x)) : !\mathbf{Set}^{\mathbf{C}^{op}}[\vec{F},\vec{G}]\times\mathbf{Set}^{!\mathbf{C}^{op}}[K,\boldsymbol{\lambda}\vec{F}]\times K\vec{C}\to\boldsymbol{\lambda}\vec{G}$$

and in the other, take $K=\boldsymbol{\lambda}\vec{F}$, $\alpha=id$, $\vec{F}=\vec{G}$, and $\beta=id$.

Summarizing, we have proved the following result:

Proposition 1 *The transformation* $\boldsymbol{\lambda}$ *defines a (pseudo)-distributive law for any choice among the six* $!_X$ *monads over the presheaf monad.*

A similar proposition is proved in [20]: there, the existence of $\boldsymbol{\lambda}$ is derived from the fact ! and *Psh* are both free constructions (the latter being the free cocompletion, i.e. the free category with all colimits) and commute in the sense that *Psh* lifts to the category of !-algebras (cf. Section 2). For completeness, we sketch this more conceptual approach in the following section.

On the other hand, the advantage of a "symbol-pushing" proof like the one presented above is to highlight plainly the "uniformity" in the choice of any combination X of Burroni's combinators.

10 Intermezzo

The reader may have noticed that the formula defining $\boldsymbol{\lambda}$ proposed in the last section "looks like" the formula for Day's tensor product (cf. section 4). In this section, we actually synthesize the latter formula from considerations of cocontinuity, and then the former one from considerations of monad lifting. Since this section has merely an explanatory purpose, we limit ourselves here to the case $! = !_\emptyset$.

Cocontinuity is at the heart of the presheaf construction. It is well-known that $\mathbf{Set}^{\mathbf{C}^{op}}$ is the free cocomplete category over $\mathbf{C}$, and that the unique cocontinuous (i.e., colimit preserving) extension of a functor $F : \mathbf{C} \to \mathbf{D}$ (where D is cocomplete, i.e., has all colimits) is its left Kan extension $Lan_{\mathcal{Y}}(F)$:

$$\begin{tikzcd} \mathbf{C} \arrow[r, "\mathcal{Y}"] \arrow[d, "F"'] & \mathbf{Set}^{\mathbf{C}^{op}} \arrow[dl, "Lan_{\mathcal{Y}}F"] \\ \mathbf{D} & \end{tikzcd}$$

In particular, the fact that $Lan_{\mathcal{Y}}(F)$ preserves colimits follows from its being left adjoint to $(C \mapsto (D \mapsto \mathbf{D}[FC, D]))$ (easy check).

We could have defined the monad *Psh* more economically and more conceptually as the monad arising from the adjunction between **Cat** and the category of cocomplete categories. The "brute force" construction of last section is in any case a good exercise in coend computations.

Although we shall not need it here, it is worth pointing out that the adjunction is monadic, i.e., that the (pseudo-) *Psh*-algebras are actually the cocomplete categories (and hence that there can be at most one *Psh*-algebra structure on a given category). This can be proved using Beck's characterization of monadic adjunctions (Theorem 4.4.4 of [6]).

Consider a slightly more general version of Day's tensor product than that given in section 4, with now some monoidal category $\mathbf{C}$ in place of $!\mathbf{1}$

$$(X \otimes Y)C = \int^{C_1,C_2} XC_1 \times XC_2 \times \mathbf{C}[C, C_1 \otimes C_2]$$

(thus Day's tensor depends on the tensor on $\mathbf{C}$). We show that this definition is entirely determined from the requirements that

- $\otimes$ extends the tensor product of $\mathbf{C}$ (via the Yoneda embedding), and
- $\otimes$ is cocontinuous in each argument.

And, of course, Day's product does satisfy these requirements (easy proof left to the reader). Recall that $\mathcal{Y}$ is dense, i.e., that every preseheaf is a colimit of representable presheaves (i.e. presheaves of the form YC), and, more precisely, that $Lan_{\mathcal{Y}}\mathcal{Y} = id$:

$$X = \int^{D} XD \cdot \mathcal{Y}D = colim_{(D, z \in XD)} \mathcal{Y}D$$

Then, we must have

$$\begin{array}{rcll}
X \otimes Y & = & (colim_{(D_1, z \in XD_1)} \mathcal{Y}D_1) \otimes Y & \\
& = & colim_{(D_1, z \in XD_1)} (\mathcal{Y}D_1 \otimes Y) & \text{(by requirement 2)} \\
& = & colim_{(D_1, z \in XD_1), (D_2, z \in XD_2)} (\mathcal{Y}D_1 \otimes \mathcal{Y}D_2) & \text{(by requirement 2)} \\
& = & colim_{(D_1, z \in XD_1), (D_2, z \in XD_2)} \mathcal{Y}(D_1 \otimes D_2) & \text{(by requirement 1)} \\
& = & \int^{D_1,D_2} (XD_1 \times XD_2) \cdot \mathcal{Y}(D_1 \otimes D_2) &
\end{array}$$

In other words, Day's tensor is determined, as announced.

It is shown in [14] that the (2-) adjunction between categories and cocomplete categories specializes to an adjunction between monoidal categories and cocomplete monoidal categories, and that $\mathbf{Set}^{\mathbf{C}^{op}}$, equipped with Day's tensor product, is the free such one. We content ourselves here with the key verification, namely that when $F : \mathbf{C} \to \mathbf{D}$ is monoidal, then so is its unique cocontinuous extension $\tilde{F} = Lan_{\mathcal{Y}}F$. We first compute $\tilde{F}X_1 \otimes \tilde{F}X_2$ and $\tilde{F}(X_1 \otimes X_2)$:

$$\tilde{F}X_1 \otimes \tilde{F}X_2 \;=\; \int^{C_1,C_2} (X_1C_1 \times X_2C_2) \cdot (FC_1 \otimes FC_2) \quad \text{(by cocontinuity of } \otimes\text{)}$$

$$\begin{array}{rcl}
\tilde{F}(X_1 \otimes X_2) & = & \int^{C} \int^{C_1,C_2} X_1C_1 \times X_2C_2 \times \mathbf{C}[C, C_1 \otimes C_2] \cdot FC \\
& = & \int^{C_1,C_2} X_1C_1 \times X_2C_2 \cdot F(C_1 \otimes C_2)
\end{array}$$

From this, it follows that the maps $FC_1 \otimes FC_2 \to F(C_1 \otimes C_2)$ induce a map $\tilde{F}X_1 \otimes \tilde{F}X_2 \to \tilde{F}(X_1 \otimes X_2)$.

As a consequence, the monad Psh on $\mathbf{Set}$ lifts to (pseudo-) !-algebras, i.e. to monoidal categories. Given two monads T, S, a lifting of S to the category of T-algebras consists of a monad S' on the category of T-algebras such that S, S' commute with the forgetful functor, and such that the unity and multiplication of S' are mapped by the forgetful functor to the unit and multiplication of S. Here, we take $S = Psh$ and $T =!$. The lifting S' maps a monoidal structure $\otimes$ on $\mathbf{C}$ to the corresponding Day's monoidal structure on $Psh(\mathbf{C})$.

We turn back to the general situation. Specifying a lifting of S to T-algebras is equivalent to giving a distributive law $TS \to ST$. This is not hard to check. Here, we just give the bit we need, namely how to define $\boldsymbol{\lambda}$ from the data of the lifting:

$$\boldsymbol{\lambda}_C = S'(\mu_C) \circ TS\eta_C$$

(recall that μ_C is a T-algebra – the free one). This allows us to derive the distributive law from Day's tensor product. First, for $S = Psh$ and $T =!$, $S'(\mu_{\mathbf{C}})$ is the Day's tensor product $(G_1, \ldots, G_n) \mapsto G_1 \otimes \ldots \otimes G_n$ associated with the (flattening) tensor product on $!\mathbf{C}$:

$$(G_1 \otimes \ldots \otimes G_n)\vec{C} = \int^{\vec{A_1},\ldots,\vec{A_n}} G_1\vec{A_1} \times \ldots \times G_n\vec{A_n} \times !\mathbf{C}[\vec{C}, (A_1^1, \ldots, A_n^{i_n})]$$

Then $\boldsymbol{\lambda}$ is obtained by replacing G_i by $Psh(\eta)(F_i)$:

$$\int^{\vec{A_1},\ldots,\vec{A_n}} (\int^{D_1} F_1D_1 \times !\mathbf{C}[\vec{A_1}, (D_1)]) \times \ldots \times (\int^{D_n} F_nD_n \times !\mathbf{C}[\vec{A_n}, (D_n)]) \times !\mathbf{C}[\vec{C}, (A_1^1, \ldots, A_n^{i_n})]$$

which simplifies to the formula given in Section 9.

11 The (bi)category $\mathbf{Prof}_?$

We next consider the coKleisli category $\mathbf{Prof}_? = (\mathbf{Cat}_{Psh})_?$. Note that the monad ! on $\mathbf{Cat}$ extends to a monad ! on $\mathbf{Prof}$, which under the self-duality of $\mathbf{Prof}$ gives a comonad ? on $\mathbf{Prof}$. Just as a monad gives rise to a Kleisli category, a comonad gives rise to a coKleisli category. Here, $\mathbf{Prof}_?$ has categories as objects, while its (1-)morphisms are defined by

$$\mathbf{Prof}_?[\mathbf{C}, \mathbf{C}'] = \mathbf{Prof}[?\mathbf{C}, \mathbf{C}']$$

For $X = \{\sigma\}$, this is the category of generalized species of structures of Fiore, Gambino, Hyland, and Winskel [11].

We are now in a position to round the circle and to show that the composition in this category on objet $\mathbf{1}$ coincides with the operation $\bullet$ defined in Section 4. We decompose all the steps of the construction.

(1) Let $\Phi : \mathbf{C} \twoheadrightarrow \mathbf{C}'$. Then $!\Phi : !\mathbf{C} \twoheadrightarrow !\mathbf{C}'$ is (using $\boldsymbol{\lambda}$):

$$!\Phi((C_1, \ldots, C_n), \vec{C'}) = \int^{D_1',\ldots,D_n'} \Phi C_1 D_1' \times \ldots \times \Phi C_n D_n' \times !\mathbf{C}'[\vec{C'}, \vec{D'}]$$

$(1)^{op}$ We can then define $?\Phi = (!\Phi^{op})^{op}$:

$$?\Phi(\vec{C},(C'_1,\ldots,C'_p) = \int^{D_1,\ldots,D_p} \Phi D_1 C'_1 \times \ldots \times \Phi D_p C'_p \times ?\mathbf{C}[\vec{D},\vec{C}]$$

(2) The multiplication of ! viewed as a monad on **Prof** at $\mathbf{C}$ is

$$(((C_1^1,\ldots,C_1^{i_1}),\ldots,(C_n^1,\ldots,C_n^{i_n})),\vec{D}) \mapsto !\mathbf{C}[\vec{D},(C_1^1,\ldots,C_n^{i_n})] : !!\mathbf{C} \twoheadrightarrow !\mathbf{C})$$

$(2)^{op}$ Hence the comultiplication of $? = (\mathbf{C} \mapsto (!(\mathbf{C}^{op})^{op})$ is

$$\delta = ((\vec{C},((D_1^1,\ldots,D_1^{i_1}),\ldots,(D_n^1,\ldots,D_n^{i_n}))) \mapsto ?\mathbf{C}[(D_1^1,\ldots,D_n^{i_n}),\vec{C}]) : ?\mathbf{C} \twoheadrightarrow ??\mathbf{C}$$

(3) Let $\Phi : ?\mathbf{C} \twoheadrightarrow \mathbf{C}'$. Then $\tilde{\Phi} = (?\Phi)\circ\delta : ?\mathbf{C} \twoheadrightarrow ?\mathbf{C}'$ is given by the following explicit formula:

$$\begin{aligned}
&\tilde{\Phi}(\vec{C},(C'_1,\ldots,C'_p)) \\
&\quad = \int^{\vec{A_1},\ldots,\vec{A_m}} ?\mathbf{C}[(A_1^1,\ldots,A_m^{j_m}),\vec{C}] \times (\int^{\vec{D_1},\ldots,\vec{D_p}} \Phi\vec{D_1}C'_1 \times \cdots \times \Phi\vec{D_p}C'_p \times ??\mathbf{C}[\vec{\vec{D}},\vec{\vec{A}}]) \\
&\quad = \int^{\vec{D_1},\ldots,\vec{D_p}} \Phi\vec{D_1}C'_1 \times \cdots \times \Phi\vec{D_p}C'_p \times ?\mathbf{C}[(D_1^1,\ldots,D_p^{i_p}),\vec{C}]
\end{aligned}$$

The simplification comes from the functoriality of the multiplication of the monad ! on **Cat** (we have $\mu_{\mathbf{C}^{op}} : !!\mathbf{C}^{op} \to !\mathbf{C}^{op}$, which we can view as a functor from $??\mathbf{C} = (!!\mathbf{C}^{op})^{op}$ to $?\mathbf{C} = (!\mathbf{C}^{op})^{op}$).

Finally, we can compose $\Phi : ?\mathbf{C} \twoheadrightarrow \mathbf{C}'$ and $\Psi : ?\mathbf{C}' \twoheadrightarrow \mathbf{C}''$: $\Psi \bullet \Phi = \Psi \circ \tilde{\Phi}$, i.e., explicitly:

$$\begin{aligned}
&(\Psi \bullet \Phi)(\vec{C},C'') \\
&\quad = \int^{C'_1,\ldots,C'_p} \Psi(\vec{C'},C'') \times \int^{\vec{D_1},\ldots,\vec{D_p}} \Phi\vec{D_1}C'_1 \times \cdots \times \Phi\vec{D_p}C'_p \times ?\mathbf{C}[(D_1^1,\ldots,D_p^{i_p}),\vec{C}]
\end{aligned}$$

As promised, we recover Kelly's (section 4) and Joyal's (section 5) settings as instances:

1. When $\mathbf{C} = \mathbf{C}' = \mathbf{C}'' = \mathbf{1}$, then Ψ and Φ are presheaves Y, X, and the definition boils down to $Y \bullet X$.

2. When $\mathbf{C} = 0$ and $\mathbf{C}' = \mathbf{C}'' = \mathbf{1}$, then Ψ is a presheaf X en Φ is a set z, and the definition boils down to $Lan_{\subseteq} X z$.

It is also easily checked that the identity I as defined in section 4 is indeed the identity of $\mathbf{Prof}_?$.

We end the section by giving a formula for $\Psi \bullet \Phi$ when ! is now an *arbitrary* monad $(!, \eta_!, \mu_!)$ over **Cat** given together with a distributive law: $\boldsymbol{\lambda} : ! \circ Psh \to Psh \circ !$. This could open the way for handling variations on the shapes of operations (the third kind of variation considered in the introduction) in a profunctor setting.

Recall (section 6) that given $\Phi : \mathbf{C} \to\!\!\!\!\!+ \mathbf{C}'$ we write Φ' for its presentation as a functor from $\mathbf{C}'$ to $\mathbf{Set}^{\mathbf{C}}$. Then we can give abstract versions of the steps $(1)^{op}$, $(2)^{op}$ and (3) above, as follows: (we still write $?\mathbf{C} = (!\mathbf{C}^{op})^{op}$):

$(1)^{op}$ Given $\Phi : \mathbf{C} \to\!\!\!\!\!+ \mathbf{C}'$, we can define $?\Phi : ?\mathbf{C} \to\!\!\!\!\!+ ?\mathbf{C}'$ by the equation:

$$(?\Phi)' = \boldsymbol{\lambda} \circ (!\Phi')$$

where on the right hand side we apply ! as a functor on $\mathbf{Cat}$.

$(2)^{op}$ The comultiplication $\delta : ?\mathbf{C} \to\!\!\!\!\!+ ??\mathbf{C}$ is defined by the formula

$$\delta' = \eta_{Psh} \circ (\mu_!)_{\mathbf{C}^{op}}$$

(3) The composition in $\mathbf{Prof}$ of $\Phi' : \mathbf{C}' \to \mathbf{Set}^{\mathbf{C}}$ and $\Psi' : \mathbf{C}'' \to \mathbf{Set}^{\mathbf{C}'}$ is defined as $\mu_{Psh} \circ Psh(\Phi') \circ \Psi'$. When $\Phi' = \eta_{Psh} \circ F$ (for some $F : \mathbf{C} \to \mathbf{C}'$), this boils down to $Psh(F) \circ \Psi'$.

Hence, given $\Phi : ?\mathbf{C} \to\!\!\!\!\!+ \mathbf{C}'$, we obtain the following formula for $\hat{\Phi} : ?\mathbf{C} \to\!\!\!\!\!+ ?\mathbf{C}'$:

$$\hat{\Phi}' = Psh(\mu_!) \circ \boldsymbol{\lambda} \circ (!\Phi')$$

Finally, expanding the definition of Psh, we get the following formula for the compostion $\Psi \bullet \Phi = \Psi \circ \hat{\Phi}$ in $\mathbf{Prof}_?$:

$$(\Psi \bullet \Phi)(\gamma, C'') = \int^{\gamma'} \Psi(\gamma', C'') \times \int^{\delta} \boldsymbol{\lambda}((!\Phi')\gamma')\delta \times ?\mathbf{C}[\mu_!\delta, \gamma]$$

Following the style of [20], one may also hide the distributive law and write

$$(\Psi \bullet \Phi)(\gamma, C'') = \int^{\gamma'} \Psi(\gamma', C'') \times \overline{\gamma'}(\Phi')\gamma$$

where $\overline{\gamma'}(\Phi')\gamma = \hat{\Phi}(\gamma, \gamma')$ can be described abstractly as follows. For each (pseudo-)!-algebra $(\mathbf{D}, \alpha : !\mathbf{D} \to \mathbf{D})$, for each category $\mathbf{C}$, and each object C of $!\mathbf{C}$, C induces a functor $\overline{C} = \mathbf{D}^{\mathbf{C}} \to \mathbf{D}$ defined as $(F \mapsto \alpha(!FC))$. The above formula is obtained by instantiating $\mathbf{C}$, C, $\mathbf{D}$ and α as $\mathbf{C}'^{op}$, γ', $\mathbf{Set}^{!(\mathbf{C}^{op})}$ and $Psh(\mu_!) \circ \boldsymbol{\lambda}$, respectively. That this is a (pseudo)-!-algebra structure is established using the equalities satisfied by $\boldsymbol{\lambda}$.

This leads to a generalized definition of operad.

Definition 1 *Given a monad* ! *on* $\mathbf{Cat}$ *and a distibutive law* $\boldsymbol{\lambda} : ! \circ Psh \to Psh \circ !$, *a* $\mathbf{C}$*-coloured* !*-operad is a monoid in* $\mathbf{Prof}_?[\mathbf{C}, \mathbf{C}]$*, that is, a functor* $X : ?\mathbf{C} \times \mathbf{C}^{op} \to \mathbf{Set}$ *given together with two natural transformations* $e : id \to X$ *and* $m : X \bullet X \to X$ *satisfying the monoid laws. A* $\mathbf{1}$*-coloured* !*-operad is called a* !*-operad.*

Hence the non symmetric operads (resp. symmetric operads, clones) are the $!_\emptyset$-operads (resp. $!_{\{\sigma\}}$-operads, $!_{\{\sigma,\delta,\epsilon\}}$-operads). But there should be many more!

12 Cooperads and properads

So far, we have addressed the variation on the "first axis" that goes from non symmetric operads to clones. In this section, we address the variation on the second axis (cooperations, bioperations).

Cooperations are dual to operations. Sets of cooperations are organized into cooperads, the notion dual to that of operad. One works now in the Kleisli category $\mathbf{Prof}_!$, i.e., a cooperad is a profunctor $X : \mathbf{1} \to\!\!\!\!\!\!+\,!\mathbf{1}$ with a monoid structure in $\mathbf{Prof}_!$,, and we can vary on the first axis as we did for operads.

For bioperations, the natural idea is to consider profunctors from $?\mathbf{1}$ to $!\mathbf{1}$. Such profunctors can compose provided there is a distributive law (another one!) $\boldsymbol{\lambda} : ?_X !_X \rightarrow !_X ?_X$: given $\Phi, \Psi : ?\mathbf{1} \to\!\!\!\!\!\!+\,!\mathbf{1}$, we define $\Psi \bullet \Phi$ by composing the comultiplication of $?$, $?\Phi$, $\boldsymbol{\lambda}$, $!\Psi$, and the multiplication of $!$. The identity is the composition ot the counit of $?$ and of the unit of $!$.

This idea has been carried out in detail by Garner, in the case of $X = \{\sigma\}$ [13]. We set

$$\boldsymbol{\lambda}((m_1, \ldots, m_p), (n_1, \ldots, n_q)) \neq \emptyset \quad \text{iff} \quad m_1 + \ldots + m_p = n_1 + \ldots + n_q$$

and when the equality holds, $\boldsymbol{\lambda}((m_1, \ldots, m_p), (n_1, \ldots, n_q))$ is the set of permutations s from $m_1 + \ldots + m_p$ to $n_1 + \ldots + n_q$ such that the graph

- whose set of vertices is the disjoint union of

  $\{1, \ldots, p\}, \{1, \ldots m_1 + \ldots + m_p\}, \{1, \ldots, n_1 + \ldots + n_q\}$, and $\{1, \ldots, q\}$

- and whose set of edges is the union
  - of the edges between 1 and $1, \ldots m_1$, between 2 and $m_1+1, \ldots, m_2$, $\ldots$, and between p and $m_{p-1} + 1, \ldots, m_p$,
  - of the edges between i and $s(i)$ ($i \leq m_1 + \ldots + m_p$), and
  - of the edges between $1, \ldots n_1$ and 1, $\ldots$, and between $n_{q-1} + 1, \ldots, n_q$ and q.

is connected. This formalizes the idea that when composing two sets of bioperations, we are interested primarily in the compositions that preserve connectedness. More precisely, composition of properations has both a vertical aspect and a horizontal aspect. The distributive law takes care of the vertical aspect. Disjoint connected compositions can be placed in parallel and composed horizontally. Thus, a full categorical account requires a double category setting, that takes care of these two aspects. We refer to [13] for details.

Monoids for this composition operation provide an analogue of operads for bioperations, and for $X = \{\sigma\}$ and $\boldsymbol{\lambda}$ as described above, what we obtain (replacing **Set** by **Vect**) is exactly the notion of *properad* introduced by Vallette in his Thèse de Doctorat [21, 22].

Acknowledgements. I collected the material presented here for an invited talk at the conference Operads 2006. When preparing this talk, I benefited a lot from discussions with Marcelo Fiore and Martin Hyland.

References

[1] M. Barr and C. Wells, Toposes, triples and theories, Springer-Verlag (1985).

[2] J. Beck, Distributive laws, Lecture Notes in Mathematics 80, 119-140 (1969).

[3] J. Bénabou, Introduction to bicategories, Lecture Notes in Mathematics 40, 1-77 (1967).

[4] J. Bénabou, Les distributeurs, Université Catholique de Louvain, Institut de Mathématique Pure et Appliquée, Rapport 33 (1973).

[5] F. Borceux, Handbook of categorical algebra, vol. I: basic category theory, Cambridge University Press (1994).

[6] F. Borceux, Handbook of categorical algebra, vol. II: categories and structures, Cambridge University Press (1994).

[7] A. Burroni, T-catégories (catégories dans un triple), Cahiers de Topologie et Géométrie Différentielle XII(3), 215-321 (1971).

[8] A. Burroni, Higher Dimensional Word Problem, Theoretical Computer Science 115, 43-62 (1993).

[9] B.J. Day, On closed categories of functors, Reports of the Midwest Category Seminar IV, Lecture Notes in Mathematics 137, 1-38 (1970).

[10] M. Fiore, Mathematical models of computational and combinatorial structures. Invited address for Foundations of Software Science and Computation Structures (FOSSACS 2005), Lecture Notes in Computer Science 3441, 25-46 (2005).

[11] M. Fiore, N.Gambino, M.Hyland, and G.Winskel, The cartesian closed bicategory of generalised species of structures. Preprint (2006).

[12] N. Gambino, On the coherence conditions for pseudo-distributive laws, draft.

[13] R. Garner, Polycategories, PhD Thesis, Cambridge University (2005).

[14] G.B. Im and G.M. Kelly, A universal property of the convolution monoidal structure, Journal of Pure and Applied Algebra 43, 75-88 (1986).

[15] A. Joyal, Foncteurs analytiques et espèces de structures, in Combinatoire énumérative, Lecture Notes in Mathematics 1234, 126-159 (1986).

[16] M. Kelly, On the operads of J.P. May, Reprints in Theory and Applications of Categories 13, 1-13 (2005).

[17] Y. Lafont, Towards an algebraic theory of Boolean circuits, Journal of Pure and Applied Algebra 184 (2-3), 257-310, Elsevier (2003).

[18] T. Leinster, Higher operads, higher categories, London Mathematical Society Lecture Note Series 298, Cambridge University Press (2004).

[19] S. Mac Lane, Categories for the working mathematician, Springer-Verlag (1971).

[20] M. Tanaka and J. Power, Pseudo-distributive laws and axiomatics for variable binding, Higher-Order and Symbolic Computation 19(2-3), 305 - 337 (2006).

[21] B. Vallette, Dualité de Koszul des PROPS, Thèse de Doctorat, Université de Strasbourg (2003).

[22] B. Vallette, A Koszul duality for props, to appear in Transactions of A.M.S.