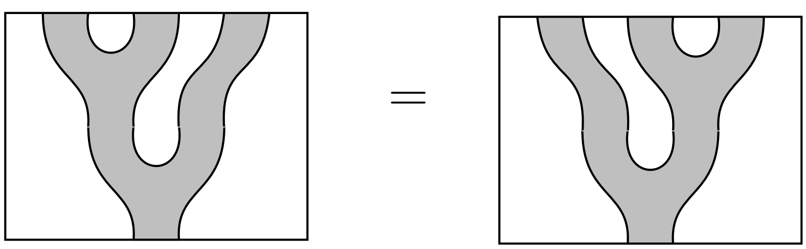
Universal Algebra and Diagrammatic Reasoning

John Baez

Geometry of Computation 2006
January 30 - Feburary 3



figures by Aaron D. Lauda

available online at
http://math.ucr.edu/home/baez/universal/

Universal Algebra

Modern universal algebra is the study of *general mathematical structures*, especially those with an 'algebraic' flavor. For example:

- *Monads* describe 'extra algebraic structure on objects of one fixed category'.
- *PROs* describe 'extra algebraic structure on objects of *any* monoidal category'.
- *PROPs* describe 'extra algebraic structure on objects of any symmetric monoidal category'.
- *Algebraic Theories* describe 'extra algebraic structure on objects of any category with finite products'.

For example:

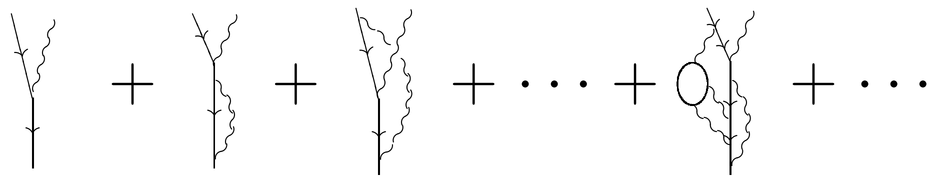
- There's a monad on Set whose algebras are groups.
- There's an algebraic theory whose algebras in any category with finite products C are 'groups in C'.
- There's no PROP whose algebras are groups, but there's a PROP for monoids.

PROPs describe fewer structures, but apply to more contexts: e.g., the category of Hilbert spaces with its tensor product. In this quantum-mechanical context we cannot 'duplicate or delete information', so the group axiom

$$g \cdot g^{-1} = 1$$

cannot be expressed.

In modern universal algebra we describe operations using diagrams with inputs and outputs. Physicists do the same with *Feynman diagrams*:



For example, all the diagrams above stand for operators between Hilbert spaces:

$$H_e \otimes H_\gamma \to H_e$$

Similar diagrams show up in other contexts:

- electrical circuits
- logic circuits
- flow charts

and part of our job is to unify these.

Monads

Using monads, we can see:

- Almost any algebraic structure has a canonical presentation in terms of generators and relations.
- But, there are ‘relations between relations’, or ‘syzygies’.
- Also relations between relations between relations, etc.
- We can build a *space* that keeps track of these: the ‘bar construction’.
- The topology of this space sheds light on the structure!

Adjunctions

We define mathematical gadgets by starting with some category A and putting extra structure on the objects of A to get objects of some fancier category B. For example:

$$\begin{array}{ll} A = \mathrm{Set} & B = \mathrm{Mon} \\ A = \mathrm{Set} & B = \mathrm{Grp} \\ A = \mathrm{AbGrp} & B = \mathrm{Ring} \\ A = \mathrm{Top} & B = \mathrm{TopGrp} \end{array}$$

In every case we have a 'forgetful' functor

$$R\colon B \to A$$

but also a 'free' functor

$$L\colon A \to B.$$

We call these **left** and **right adjoints** if there is a natural isomorphism

$$\hom(La, b) \cong \hom(a, Rb).$$

The Canonical Presentation

Given an adjunction

$$A \underset{R}{\overset{L}{\rightleftarrows}} B$$

let's try to get a 'presentation' of $b \in B$. First note:

$$\begin{aligned} \hom(LRb, b) &\cong \hom(Rb, Rb) \\ e_b &\mapsto 1_{Rb} \end{aligned}$$

for some morphism

$$e_b \colon LRb \to b$$

called the **counit**.

In the example

$$A = \text{Set} \quad B = \text{Mon}$$

the monoid LRb consists of words of elements of the monoid b. The counit maps 'formal products' in LRb to actual products in b.

So, we have the raw material for a presentation! To see the *relations*, form:

$$LRLRb \underset{LR(e_b)}{\overset{e_{LRb}}{\rightrightarrows}} LRb \xrightarrow{\quad e_b \quad} b$$

This diagram always commutes. It's enlightening to check this when

$$A = \mathrm{Set} \quad B = \mathrm{Mon}.$$

Then $LRLRb$ consists of 'words of words', and the commuting diagam above says these give *relations* in the presentation of b where all elements of b are generators.

In this example the diagram is a 'coequalizer', so we really have a presentation. This is not always true — but it is when the adjunction is 'monadic'.

Relations Between Relations

The canonical presentation is highly redundant, so there will be relations between relations. We can see these by forming a **resolution** of b:

$$\cdots \substack{\longrightarrow\\ \longrightarrow\\ \longrightarrow\\ \longrightarrow} (LR)^3 b \substack{\longrightarrow\\ \longrightarrow\\ \longrightarrow} (LR)^2 b \substack{\longrightarrow\\ \longrightarrow} LRb \longrightarrow b$$

When objects of B are sets with extra structure, this gives a 'simplicial set' with:

- LRb as vertices,
- $(LR)^2 b$ as edges,
- $(LR)^3 b$ as triangles....

In general we get a 'simplicial object in B'. This is called the **bar construction**.

Exercise: work out the details when

$$A = \mathrm{Set} \quad B = \mathrm{Mon}.$$

Unit and Counit

Just as

$$\begin{aligned} \hom(LRb, b) &\cong \hom(Rb, Rb) \\ e_b &\mapsto 1_{Rb} \end{aligned}$$

gives the **counit**

$$e_b \colon LRb \to b$$

which 'evaluates formal expressions', so

$$\begin{aligned} \hom(La, La) &\cong \hom(a, RLa) \\ 1_{La} &\mapsto i_a \end{aligned}$$

gives the **unit**

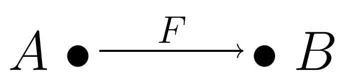
$$i_a \colon a \to RLa$$

which 'maps generators into the free algebra'. The unit and counit satisfy certain identities, best written as diagrams....

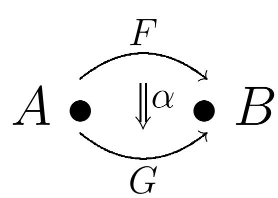
2-Categories

We've been talking about categories:

functors:



and natural transformations:



These are the objects, morphisms and 2-morphisms of the 2-category Cat. The definition of '2-category' summarizes what we can do with them in a purely diagrammatic way!

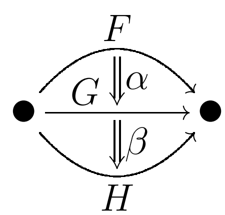
In a **2-category**:

• we can compose morphisms:

$$\bullet \xrightarrow{F} \bullet \xrightarrow{G} \bullet$$

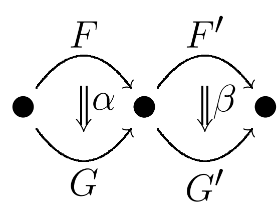
with associativity and identities 1_A

• we can compose 2-morphisms **vertically**:



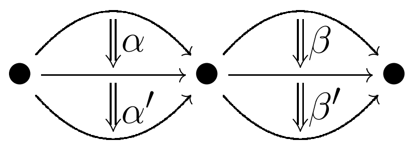
with associativity and identities 1_F

• we can compose 2-morphisms **horizontally**:



with associativity and identities 1_{1_A}

• and finally, the two ways of parsing this agree:



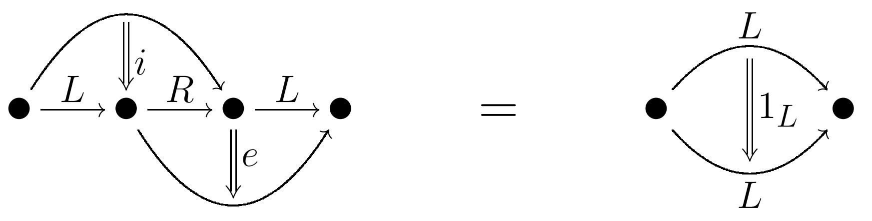
In other words:

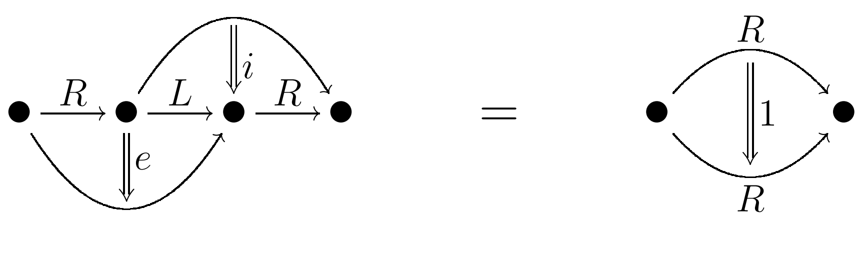
$$(\beta'\beta) \circ (\alpha'\alpha) = (\beta' \circ \alpha')(\beta \circ \alpha)$$

In this notation, an **adjunction** consists of $A \underset{R}{\overset{L}{\rightleftarrows}} B$ with

$$i\colon 1_A \Rightarrow RL, \quad e\colon LR \Rightarrow 1_B$$

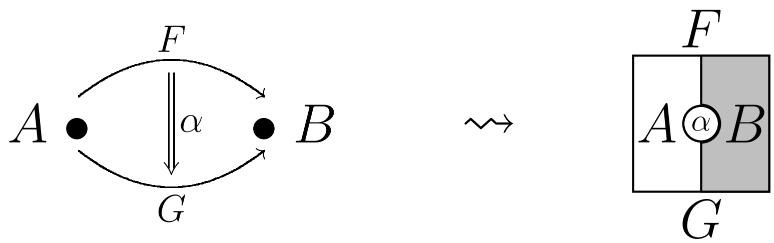
satisfying the **zig-zag identities**:





String Diagrams

It's fun to convert the above 'globular' diagrams into string diagrams:

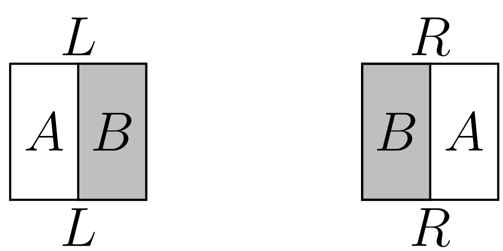


in which objects of a 2-category become 2d regions:

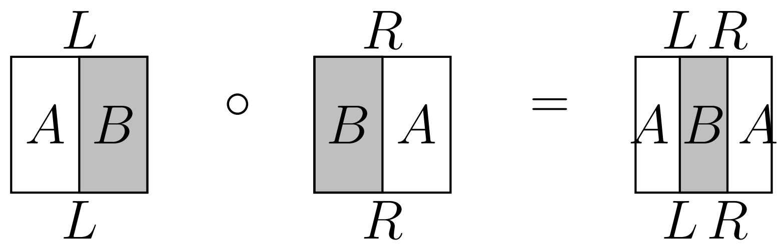
morphisms become 1d edges, and 2-morphisms become 0d vertices thickened to discs. For example,

$$L: A \to B, \qquad R: B \to A$$

are drawn as:



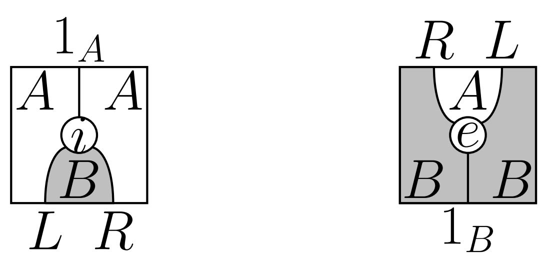
$RL\colon A \to A$ is drawn as:



The unit and counit

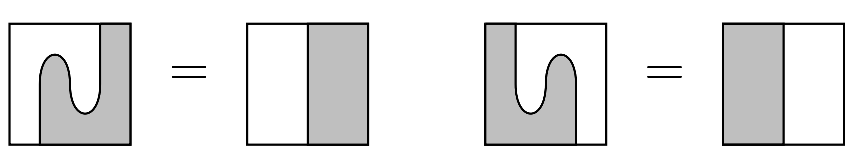
$$i\colon 1 \Rightarrow RL, \quad e\colon LR \Rightarrow 1$$

look like this:



or for short:

The zig-zag identities become:



From the definition of adjunction we can derive properties of

$$M = RL\colon A \to A$$

and

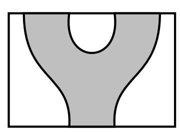
$$C = LR\colon B \to B$$

and these become the definitions of a **monad** and **comonad**.

For M we get a **multiplication**

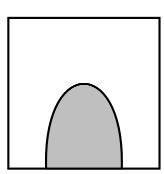
$$m\colon M^2 \Rightarrow M$$

defined by:

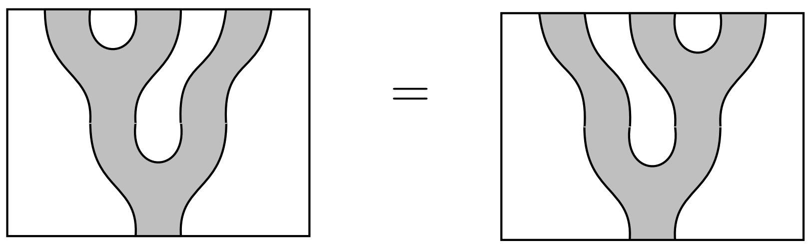


We also have the **unit**

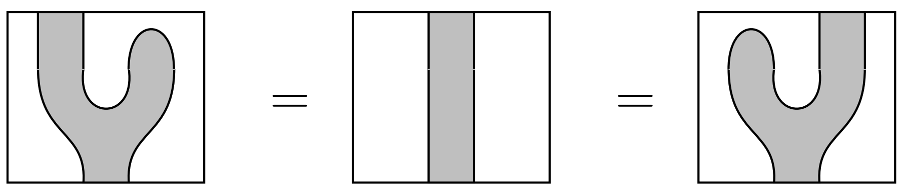
$$i\colon 1 \Rightarrow M$$



These satisfy **associativity**:



and the **left and right unit laws**:



These are 'topologically true': the zig-zag identity and 2-category axioms let us prove them by warping the pictures as if drawn on rubber!

Similarly, a **comonad** has a **comultiplication** and **counit** satisfying **coassociativity** and the **left and right counit laws**. To draw all these, just turn the above pictures upside down and switch white and shaded regions.

The Bar Construction

Suppose $M\colon A \to A$ is a monad. Let Δ be the category of finite ordinals and order-preserving maps. For any order-preserving map

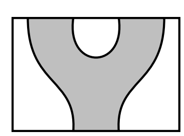
$$f\colon [i] \to [j],$$

our pictures give us a 2-morphism

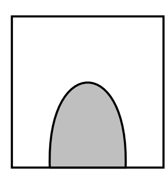
$$F(f)\colon M^i \Rightarrow M^j$$

built from the multiplication and unit:

$$\begin{array}{c} 1+1 \\ \downarrow \\ 1 \end{array} \quad \rightsquigarrow$$



$$\begin{array}{c} 0 \\ \downarrow \\ 1 \end{array} \quad \rightsquigarrow$$



So, we get a functor

$$F\colon \Delta \to \hom(A, A).$$

If $C\colon B \to B$ is a comonad, the same pictures upside-down with colors switched give a functor

$$F\colon \Delta^{\mathrm{op}} \to \hom(B, B).$$

A functor from Δ^{op} is called an **augmented simplicial object**. In particular, if C is a comonad in Cat and $b \in B$ is an object, we get

$$F(b)\colon \Delta^{\mathrm{op}} \to B,$$

an augmented simplicial object in B:

$$\cdots \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} C^3b \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} C^2b \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} Cb \longrightarrow b$$

This portion is called **the bar construction**:

$$\bar{b} = \Big\{ \cdots \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} C^3b \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} C^2b \begin{smallmatrix}\longrightarrow\\ \longleftarrow\\ \longrightarrow\end{smallmatrix} Cb \Big\}$$

$\bar{b}$ is a **simplicial object**: a functor from the opposite of the category of *nonempty* finite ordinals.

The Moral: whenever we have a comonad $C\colon B \to B$, the bar construction lets us 'puff up' any object $b \in B$ to a simplicial object $\bar{b}$ in which:

- formal products of generators become vertices
- relations become edges
- relations between relations become triangles, etc....

Paths in $\bar{b}$ are 'rewrites', 'proofs' or 'computations' using relations in the canonical presentation of b — *and we can define homotopies between paths, etc!*

So, we can study rewriting processes of rewriting processes of rewriting... to our heart's content.

Algebras of Monads

To state good theorems about the bar construction, we must restrict to 'monadic' adjunctions.

We know every adjunction gives a monad. Next we'll show every monad

$$M\colon A \to A$$

gives an adjunction

$$A \underset{R}{\overset{L}{\rightleftarrows}} A^M$$

where A^M is the 'category of algebras of M'. An adjunction is **monadic** if it arises this way.

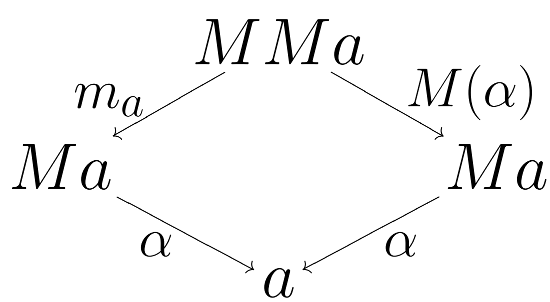
Consider the 'monad for monoids'

$$M\colon \mathrm{Set} \to \mathrm{Set}$$

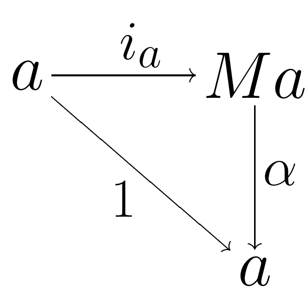
sending every set a to the set Ma of words in a. To make a set a into a monoid, we should choose

$$\alpha\colon Ma \to a$$

mapping formal products to actual products in a. This should be **associative**:



and obey the **left unit law**:



In general, given any monad

$$M\colon A \to A,$$

we call a $\alpha\colon Ma \to a$ an **algebra of** M if it obeys associativity and the left unit law.

There's a **category of algebras of** M, the **Eilenberg–Moore category** A^M. And, there's an adjunction

$$A \underset{R}{\overset{L}{\rightleftarrows}} A^M$$

All these adjunctions:

$$\begin{array}{ll} A = \mathrm{Set} & A^M = \mathrm{Mon} \\ A = \mathrm{Set} & A^M = \mathrm{Grp} \\ A = \mathrm{AbGrp} & A^M = \mathrm{Ring} \\ A = \mathrm{Top} & A^M = \mathrm{TopGrp} \end{array}$$

arise from monads this way, so we call them **monadic**.

The Big Theorem

Suppose

$$A \underset{R}{\overset{L}{\rightleftarrows}} B$$

is a monadic adjunction and $C = LR$ the resulting comonad. Then any object b is a coequalizer of its canonical presentation:

$$C^2 b \underset{C(e_b)}{\overset{e_{Cb}}{\rightrightarrows}} Cb \xrightarrow{e_b} b$$

Moreover,

$$\bar{b} = \left\{ \cdots \ C^3 b \ C^2 b \rightrightarrows Cb \right\}$$

is the initial **b-acyclic** simplicial object in B — namely, the initial one for which $R\bar{b}$ is equipped with a deformation retraction to Rb.

First part: see Beck's monadicity theorem. Second part: see Todd Trimble's notes (in the references).

Some Remarks

1. We can get subobjects of $\bar{b}$ using more efficient choices of generators, or relations, or relations between relations.... This is where Squier's work on confluent terminating presentations comes in!

2. A b-acyclic simplicial object in B is sometimes called a **resolution** of b. This terminology is most widespread in 'homological algebra', which studies simplicial abelian groups, otherwise known as **chain complexes** of abelian groups.

Strict Monoidal Categories

A 2-category with one object is called a **strict monoidal category**:

Monoidal Category	**2-Category**
•	objects
objects	morphisms
morphisms	2-morphisms
tensor product of objects	composite of morphisms
composite of morphisms	vertical composite of 2-morphisms
tensor product of morphisms	horizontal composite of 2-morphisms

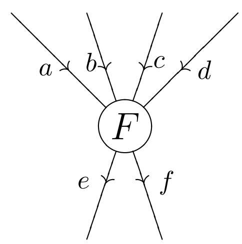
For example: Set with its Cartesian product $\times$, or Vect with its tensor product $\otimes$, cleverly made to be strictly associative and unital.

Feynman Diagrams and Tensors

Physicists draw morphisms in monoidal categories:

$$F\colon a \otimes b \otimes c \otimes d \to e \otimes f$$

either as **Feynman diagrams**:

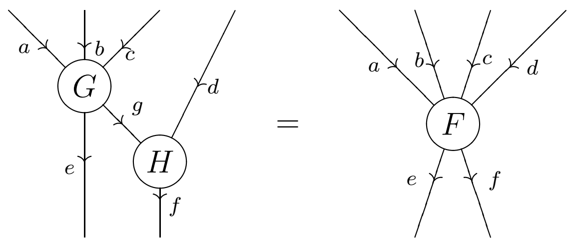


which are just string diagrams with no shading on regions, or as **tensors**:

$$F^{abcd}_{ef}$$

with indices standing for inputs and outputs. Penrose called the latter approach **abstract index notation**.

In a monoidal category, we can compose/tensor morphisms like this:



or equivalently, like this:

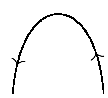
$$G^{abc}_{eg} H^{gd}_{f} = F^{abcd}_{ef}$$

where the **Einstein summation convention** says that any index appearing once as a superscript and once as a subscript — g in this example — labels an 'internal edge'.

In particle physics, edges of Feynman diagrams are called **particles**. Vertices are called **interactions**. Internal edges are called **virtual particles**.

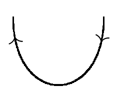
Monads in Vect

In Vect, an *adjunction* turns out to be a pair of dual vector spaces V, V^* with the unit



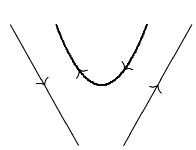
$$\begin{array}{c} k \\ \big\downarrow i \\ V \otimes V^* \end{array}$$

and counit



$$\begin{array}{c} V^* \otimes V \\ \big\downarrow e \\ k \end{array}$$

being the obvious maps. A *monad* in Vect is an associative algebra. Since an adjunction gives a monad, a pair of dual vector spaces gives an associative algebra: the matrix algebra $M = V \otimes V^*$.

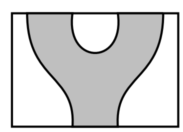


$$\begin{array}{c} M \otimes M \\ \big\downarrow m \\ M \end{array}$$

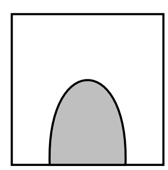
A Monad in Δ

The category of finite ordinals, Δ, is a strict monoidal category with $+$ as its tensor product. $1 \in \Delta$ is a monad with multiplication and unit:

$$\begin{array}{c} 1+1 \\ \big\downarrow{\scriptstyle m} \\ 1 \end{array}$$



$$\begin{array}{c} 0 \\ \big\downarrow{\scriptstyle i} \\ 1 \end{array}$$



People usually call a monad in a strict monoidal category a **monoid object**, so let's do that! For example: a monoid object in Vect is an associative algebra.

Δ is special because it's the *free monoidal category on a monoid object!* In other words...

Theorem: Given a strict monoidal category $\mathcal{C}$, there's a 1-1 correspondence between monoid objects in $\mathcal{C}$ and functors

$$F\colon \Delta \to \mathcal{C}$$

that preserve tensor products. Each such functor gives a monoid object $F(1)$ with multiplication $F(m)$ and unit $F(i)$.

This explains the special role of Δ in our discussion of monads. A monad $M\colon A \to A$ is just a monoid object in $\hom(A, A)$, so it gives us

$$F\colon \Delta \to \hom(A, A).$$

Similarly: a comonad $C\colon B \to B$ is a monoid object in $\hom(B, B)^{\mathrm{op}}$, so we get

$$F\colon \Delta^{\mathrm{op}} \to \hom(B, B).$$

Monoidal Categories

Alas, strict monoidal categories are rare in nature. In general, a **monoidal category** is:

- a category $\mathcal{C}$
- a functor $\otimes \colon \mathcal{C} \times \mathcal{C} \to \mathcal{C}$
- a **unit object** $I \in \mathcal{C}$
- natural isomorphisms called the **associator**:

  $a_{x,y,z} \colon (x \otimes y) \otimes z \to x \otimes (y \otimes z)$

  left unitor:

  $$\ell_x \colon I \otimes x \to x$$

  and **right unitor**:

  $$r_x \colon x \otimes I \to x$$

such that...

the following diagrams commute:

- the **pentagon equation**:

$$
\begin{array}{ccc}
 & (w \otimes x) \otimes (y \otimes z) & \\
 {\scriptstyle a_{w \otimes x, y, z}} \nearrow & & \searrow {\scriptstyle a_{w, x, y \otimes z}} \\
((w \otimes x) \otimes y) \otimes z & & w \otimes (x \otimes (y \otimes z)) \\
{\scriptstyle a_{w,x,y} \otimes 1} \downarrow & & \uparrow {\scriptstyle 1 \otimes a_{x,y,z}} \\
(w \otimes (x \otimes y)) \otimes z & \xrightarrow{a_{w, x \otimes y, z}} & w \otimes ((x \otimes y) \otimes z)
\end{array}
$$

- the **triangle equation**:

$$
\begin{array}{ccc}
(x \otimes I) \otimes y & \xrightarrow{a_{x,I,y}} & x \otimes (I \otimes y) \\
 {\scriptstyle r_x \otimes I} \searrow & & \swarrow {\scriptstyle 1 \otimes \ell_y} \\
 & x \otimes y &
\end{array}
$$

Mac Lane's coherence theorem says these laws suffice to make all diagrams built from $\otimes, a, \ell, r$ commute.

A functor $F: \mathcal{C} \to \mathcal{D}$ between monoidal categories is **monoidal** if it is equipped with:

- a natural isomorphism $\Phi_{x,y}: F(x) \otimes F(y) \to F(x \otimes y)$
- an isomorphism $\phi: I \to F(I)$

such that these diagrams commute:

- **compatibility with unitors**:

$$
\begin{array}{ccc}
I \otimes F(x) & \xrightarrow{\ell_{F(x)}} & F(x) \\
\Big\downarrow {\scriptstyle \phi \otimes 1} & & \Big\uparrow {\scriptstyle F(\ell_x)} \\
F(I) \otimes F(x) & \xrightarrow[\Phi_{I,x}]{} & F(I \otimes x)
\end{array}
$$

$$
\begin{array}{ccc}
F(x) \otimes I & \xrightarrow{r_{F(x)}} & F(x) \\
\Big\downarrow {\scriptstyle 1 \otimes \phi} & & \Big\uparrow {\scriptstyle F(r_x)} \\
F(x) \otimes F(I) & \xrightarrow[\Phi_{x,I}]{} & F(x \otimes I)
\end{array}
$$

- **compatibility with the associator**:

$$\begin{array}{ccc}
(F(x)\otimes F(y))\otimes F(z) & \xrightarrow{a_{F(x),F(y),F(z)}} & F(x)\otimes(F(y)\otimes F(z)) \\
\Big\downarrow{\scriptstyle \Phi_{x,y}\otimes 1} & & \Big\downarrow{\scriptstyle 1\otimes\Phi_{y,z}} \\
F(x\otimes y)\otimes F(z) & & F(x)\otimes F(y\otimes z) \\
\Big\downarrow{\scriptstyle \Phi_{x\otimes y,z}} & & \Big\downarrow{\scriptstyle \Phi_{x,y\otimes z}} \\
F((x\otimes y)\otimes z) & \xrightarrow[F(a_{x,y,z})]{} & F(x\otimes(y\otimes z))
\end{array}$$

Exercise: given a monoid object M in $\mathcal{C}$ and a monoidal functor

$$F\colon \mathcal{C}\to\mathcal{D},$$

check that $F(M)$ becomes a monoid object in $\mathcal{D}$.

PROs

A **PRO** is a monoidal category $\mathcal{C}$ for which every object equals $x^{\otimes n}$ for some fixed object $x \in \mathcal{C}$. Given monoidal categories $\mathcal{C}$ and $\mathcal{D}$, an **algebra of $\mathcal{C}$ in $\mathcal{D}$** is a monoidal functor

$$F: \mathcal{C} \to \mathcal{D}$$

If $\mathcal{C}$ is a PRO, this picks out an object $F(x) \in \mathcal{D}$ and equips it with algebraic structure.

For example, Δ is a PRO since every object is of the form $1 + \cdots + 1$. Δ is called the *PRO for monoids*, since an algebra of Δ in $\mathcal{D}$

$$F: \Delta \to \mathcal{D}$$

picks out $F(1) \in \mathcal{D}$ and equips it with the structure of a monoid object.

There's a PRO for any untyped algebraic structure with operations of arbitrary source and target arity satisfying equational laws with *no duplication, deletion or permutation* of arguments. To eliminate the last restriction, we need 'PROPs'. To eliminate all three, we need 'algebraic theories':

$$\begin{array}{ccc}
\mathrm{AlgTh} & \hookrightarrow & \mathrm{FinProdCat} \\
\downarrow & & \downarrow \\
\mathrm{PROP} & \hookrightarrow & \mathrm{SymmMonCat} \\
\downarrow & & \downarrow \\
\mathrm{PRO} & \hookrightarrow & \mathrm{MonCat}
\end{array}$$

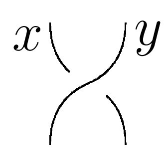
Gadgets of the right-hand sort describe *typed* algebraic structures.

PROPs

There's no PRO for commutative monoids; to say $ab = ba$ we need to permute arguments. For this we need PROPs, which are special symmetric monoidal categories.

A **braided monoidal category** $\mathcal{C}$ is a monoidal category equipped with a natural isomorphism called the **braiding**:

$$B_{x,y}: x \otimes y \to y \otimes x$$



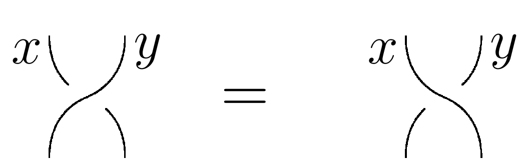
making these diagrams commute...

- the **hexagon equations**:

$$
\begin{array}{ccccc}
x \otimes (y \otimes z) & \xrightarrow{a^{-1}_{x,y,z}} & (x \otimes y) \otimes z & \xrightarrow{B_{x,y} \otimes z} & (y \otimes x) \otimes z \\
\Big\downarrow {\scriptstyle B_{x,y\otimes z}} & & & & \Big\downarrow {\scriptstyle a_{y,x,z}} \\
(y \otimes z) \otimes x & \xrightarrow[a_{y,z,x}]{} & y \otimes (z \otimes x) & \xrightarrow[y \otimes B_{x,z}]{} & y \otimes (x \otimes z)
\end{array}
$$

$$
\begin{array}{ccccc}
(x \otimes y) \otimes z & \xrightarrow{a_{x,y,z}} & x \otimes (y \otimes z) & \xrightarrow{x \otimes B_{y,z}} & x \otimes (z \otimes y) \\
\Big\downarrow {\scriptstyle B_{x\otimes y,z}} & & & & \Big\downarrow {\scriptstyle a^{-1}_{x,z,y}} \\
z \otimes (x \otimes y) & \xrightarrow[a^{-1}_{z,x,y}]{} & (z \otimes x) \otimes y & \xrightarrow[B_{x,z} \otimes y]{} & (x \otimes z) \otimes y
\end{array}
$$

A **symmetric monoidal category** is a braided monoidal category with $B_{x,y} = B^{-1}_{y,x}$:



For example: (Set, $\times$), (Set, $+$), or (Vect, $\otimes$), but not Δ.

A functor $F: \mathcal{C} \to \mathcal{D}$ between symmetric monoidal categories is **symmetric monoidal** if it's monoidal and satisfies:

- **compatibility with the braiding**:

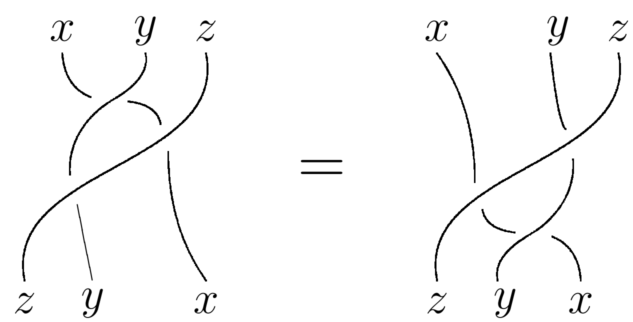
$$\begin{array}{ccc} F(x) \otimes F(y) & \xrightarrow{B_{F(x),F(y)}} & F(y) \otimes F(x) \\ \downarrow{\scriptstyle \Phi_{x,y}} & & \downarrow{\scriptstyle \Phi_{y,x}} \\ F(x \otimes y) & \xrightarrow[F(B_{x,y})]{} & F(y \otimes x) \end{array}$$

A **PROP** is a symmetric monoidal category $\mathcal{C}$ for which every object equals $x^{\otimes n}$ for some fixed object $x \in \mathcal{C}$. Given symmetric monoidal categories $\mathcal{C}$ and $\mathcal{D}$, an **algebra of $\mathcal{C}$ in $\mathcal{D}$** is a symmetric monoidal functor

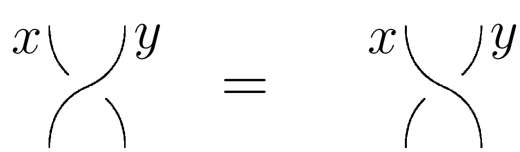
$$F: \mathcal{C} \to \mathcal{D}$$

String Diagrams Revisited

We can reason in monoidal categories using 2-dimensional string diagrams. Similarly, we can reason in braided monoidal categories using 3-dimensional string diagrams:



We can reason in symmetric monoidal categories using 4-dimensional (or higher-dimensional) string diagrams, since now:



Exercise: describe the PROP for commutative monoids using string diagrams.

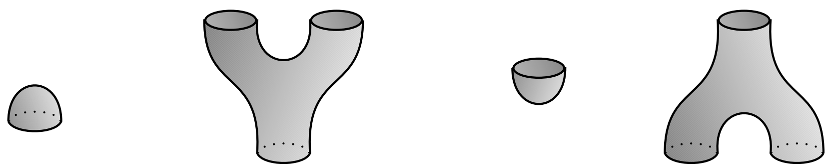
Examples

The category FinSet of finite sets and functions is a PROP with $+$ as its tensor product, since every object equals $1 + \cdots + 1$. And this is the *PROP for commutative monoids!*

In other words, $(\mathrm{FinSet}, +)$ is the *free symmetric monoidal category on a commutative monoid object*, just as Δ is the free monoidal category on a monoid object.

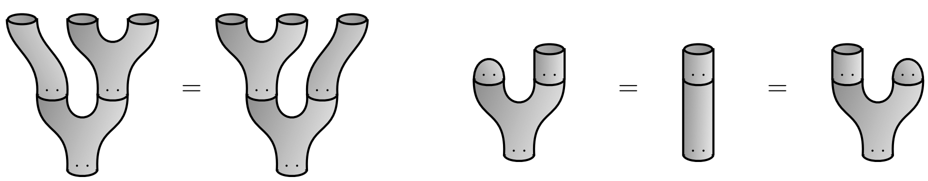
Δ^{op} is the *PRO for comonoids*, since a monoidal functor $F\colon \Delta^{\mathrm{op}} \to \mathcal{D}$ is the same as a monoid object in $\mathcal{D}^{\mathrm{op}}$ — which we call a **comonoid object** in $\mathcal{D}$. Similarly, $(\mathrm{FinSet}^{\mathrm{op}}, +)$ is the *PROP for cocommutative comonoids.*

In physics, it's important that the category of 2d cobordisms is the *PROP for commutative Frobenius algebras*:

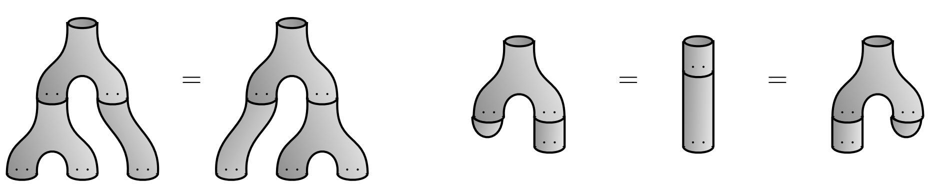


$i\colon I \to x$ $\quad m\colon x \otimes x \to x$ $\quad e\colon x \to I$ $\quad \Delta\colon x \to x \otimes x$

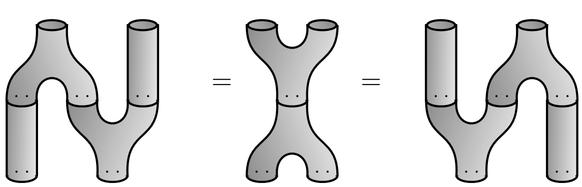
Monoid laws:

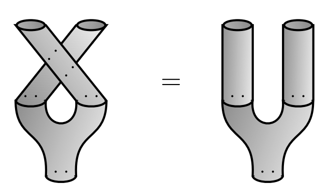


Comonoid laws:

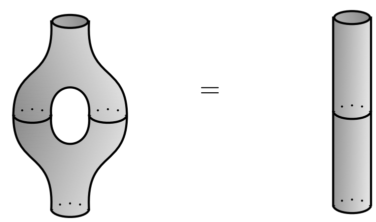


Frobenius laws: Commutativity:





Any semisimple commutative algebra naturally gives rise to a commutative Frobenius algebra satisfying this extra law:



Commutative Frobenius algebras of this sort are called *strongly separable*. The PROP for these algebras is the category of cospans of finite sets:

$$X \xrightarrow{f} A \xleftarrow{g} Y$$

See Rosebrugh, Sabadini and Walters for a proof. For Frobenius algebras in physics, see Lauda and Pfeiffer. (Links on my webpage.)

Algebraic Theories

A category $\mathcal{C}$ has **finite products** if it has:

• a **terminal object** 1, meaning that for each $x \in \mathcal{C}$ there is a unique morphism

$$x \overset{!}{\dashrightarrow} 1$$

• for each pair of objects x, y a **binary product**

$$\begin{array}{ccc} & x \times y & \\ {}^{\pi_x}\swarrow & & \searrow^{\pi_y} \\ x & & y \end{array}$$

meaning that any pair of maps to x and y factors uniquely through $x \times y$:

$$\begin{array}{ccccc} & & a & & \\ & {}^{f}\swarrow & \big\downarrow{!} & \searrow^{g} & \\ & & x \times y & & \\ x & \overset{\pi_x}{\longleftarrow} & & \overset{\pi_y}{\longrightarrow} & y \end{array}$$

An **algebraic theory** is a category with finite products such that every object equals x^n for some fixed object x. Given categories with finite products $\mathcal{C}$ and $\mathcal{D}$, an **algebra of $\mathcal{C}$ in $\mathcal{D}$** is a product preserving functor

$$F\colon \mathcal{C} \to \mathcal{D}$$

Algebraic theories can be seen as special PROPs, since any category with finite products becomes a symmetric monoidal category if we take $\otimes$ to be $\times$ and I to be 1.

(There's usually a *choice* of binary products and terminal object, since they're only unique up to canonical isomorphism. But this is harmless.)

Theorem: A symmetric monoidal category gives a category with finite products with $\times = \otimes$, $1 = I$ iff there are monoidal natural transformations

$$e_x \colon x \to I$$

and

$$\Delta_x \colon x \to x \otimes x$$

such that these commute:

$$\begin{array}{ccc} x & \xrightarrow{\Delta_x} & x \otimes x \\ {\scriptstyle 1}\downarrow & & \downarrow{\scriptstyle e_x \otimes 1} \\ x & \xleftarrow{\ell_x} & I \otimes x \end{array} \qquad \begin{array}{ccc} x & \xrightarrow{\Delta_x} & x \otimes x \\ {\scriptstyle 1}\downarrow & & \downarrow{\scriptstyle 1 \otimes e_x} \\ x & \xleftarrow{r_x} & x \otimes I \end{array}$$

So, a symmetric monoidal category becomes a category with finite products when we can *duplicate and delete variables* using

$$\Delta_x \colon x \to x \otimes x, \quad e_x \colon x \to I$$

Exercise: Say this using string diagrams.

Examples

There's an algebraic theory for any untyped algebraic structure with operations of arbitrary (finite) arity satisfying equational laws. For example:

- groups
- rings
- *not fields!*
- vector spaces over a given field k
- associative algebras over k

Using categories with finite products we can describe any *typed* algebraic structure with operations of arbitrary arity satisfying equational laws. For example:

- chain complexes of abelian groups
- ring/module pairs

New Versus Old

Lawvere invented algebraic theories to streamline Birkhoff's approach to universal algebra where a **variety** is a set of operations, e.g.:

$\cdot$ (binary) $^{-1}$ (unary) 1 (nullary)

together with a set of purely equational axioms, e.g.:

$$(g\cdot h)\cdot k = g\cdot(h\cdot k), \quad 1\cdot g = 1, \quad g\cdot 1 = g,$$
$$g\cdot g^{-1} = 1, \quad g^{-1}\cdot g = 1.$$

More generally, categories with finite products $\mathcal{C}$ do the same job as 'typed varieties'. Given $\mathcal{C}$ we get a typed variety with:

- one type per object of $\mathcal{C}$
- one operation per morphism
- one axiom per equation between morphisms

From this viewpoint, we can think of PROPs as special algebraic theories where no variable appears twice on the same side of any equation, and the same variables appear on both sides. For example,

$$(g \cdot h) \cdot k = g \cdot (h \cdot k)$$

is okay, but not

$$g \cdot g = g$$

or

$$g \cdot h = g.$$

In short: *no duplication or deletion of variables!*

This is related to the fact that 'we cannot clone a quantum', nor can we 'delete a quantum': $(\mathrm{Hilb}, \otimes)$ is a symmetric monoidal category where $\otimes$ is not the binary product and 1 is not terminal.

Q: Viewed as lists of axioms, PROPs are special algebraic theories. Viewed as categories, algebraic theories are special PROPs! *What's going on?*

A: There's a kind of 'adjunction':

$$\mathrm{PROP} \underset{R}{\overset{L}{\rightleftarrows}} \mathrm{AlgTh}$$

or more generally

$$\mathrm{SymmMonCat} \underset{R}{\overset{L}{\rightleftarrows}} \mathrm{FinProdCat}$$

since

$$\hom(L\mathcal{C}, \mathcal{D}) \simeq \hom(\mathcal{C}, R\mathcal{D})$$

in a 'natural' way. For example: think of the PROP for monoids $\mathcal{C}$ as a special algebraic theory $L\mathcal{C}$, and look at algebras of $L\mathcal{C}$ in some category with finite products $\mathcal{D}$. These are the same as algebras of $\mathcal{C}$ in the underlying symmetric monoidal category $R\mathcal{D}$ of $\mathcal{D}$!

In fact

$$L\colon \text{SymmMonCat} \to \text{FinProdCat}$$

and

$$R\colon \text{FinProdCat} \to \text{SymmMonCat}$$

are '2-functors' between 2-categories, and the 'adjunction' between them is actually a 'pseudo-adjunction', since we only have an equivalence of categories

$$\hom(L\mathcal{C}, \mathcal{D}) \simeq \hom(\mathcal{C}, R\mathcal{D})$$

instead of an isomorphism.

Instead of studying these general concepts, let's just study L and R. They're interesting because they go between the 'classical' world of finite product categories (e.g. algebraic theories) and the 'quantum' world of symmetric monoidal categories (e.g. PROPs).

Q: What does

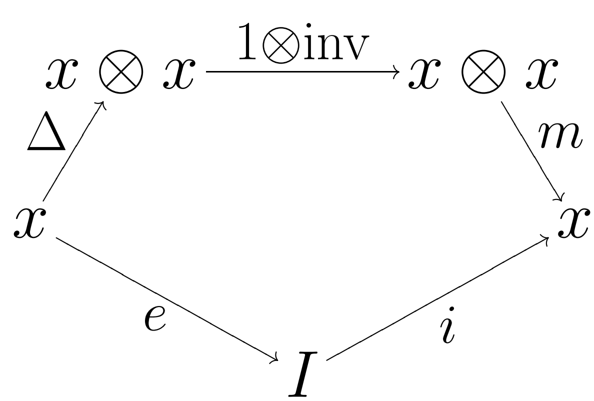
$$R\colon \mathrm{FinProdCat} \to \mathrm{SymmMonCat}$$

do when applied to an algebraic theory like the theory of groups? What sort of PROP do we get?

A: We get a PROP in which the diagonal and map to the unit object have become explicit operations

$$\Delta\colon x \to x \otimes x, \quad e\colon x \to I$$

For example, if $\mathcal{D}$ is the algebraic theory for groups, $R\mathcal{D}$ is the PROP for **cocommutative Hopf algebras**. In here the axiom $g{\cdot}g^{-1} = 1$ becomes:



Q: What does the 'pseudo-monad'

$$\mathrm{SymmMonCat} \xrightarrow{RL} \mathrm{SymmMonCat}$$

do?

A: Given a symmetric monoidal category $\mathcal{C}$, $M = RL$ throws in new morphisms

$$\Delta_x \colon x \to x \otimes x, \quad e_x \colon x \to I$$

satisfying the conditions that make $M\mathcal{C}$ into a category with finite products. In other words, *M gives $\mathcal{C}$ the ability to duplicate and delete information!*

We will describe M in a beautiful way using the 'tensor product' of symmetric monoidal categories.

SymmMonCat is a 2-category with:

- symmetric monoidal categories as objects
- symmetric monoidal functors as morphisms
- 'monoidal natural transformations' as 2-morphisms

where a natural transformation

$$\mathcal{C} \bullet \underset{G}{\overset{F}{\Downarrow \alpha}} \bullet \mathcal{D}$$

between monoidal functors is **monoidal** if it gets along with tensor products in the following way...

- **compatibility with** $\otimes$:

$$\begin{array}{ccc} F(x)\otimes F(y) & \xrightarrow{\alpha_x\otimes\alpha_y} & G(x)\otimes G(y) \\ \Big\downarrow{\scriptstyle \Phi_{x,y}} & & \Big\downarrow{\scriptstyle \Gamma_{x,y}} \\ F(x\otimes y) & \xrightarrow[\alpha_{x\otimes y}]{} & G(x\otimes y) \end{array}$$

- **compatibility with** I:

$$\begin{array}{ccc} & I & \\ {\scriptstyle \phi}\swarrow & & \searrow{\scriptstyle \gamma} \\ F(I) & \xrightarrow[\alpha_I]{} & G(I) \end{array}$$

Here the isomorphisms

$$\Phi_{x,y}\colon F(x)\otimes F(y)\to F(x\otimes y)$$

and

$$\phi\colon I\to F(I)$$

are part of the monoidal functor F, and similarly Γ and γ are part of G.

So, given symmetric monoidal categories $\mathcal{C}$ and $\mathcal{D}$, we get a category $\hom(\mathcal{C}, \mathcal{D})$ with:

- symmetric monoidal functors $F: \mathcal{C} \to \mathcal{D}$ as objects
- monoidal natural transformations $\alpha: F \to G$ as morphisms

or in other words:

- **algebras of $\mathcal{C}$ in $\mathcal{D}$** as objects
- **algebra homomorphisms** as morphisms.

We call $\hom(\mathcal{C}, \mathcal{D})$ the **category of algebras of $\mathcal{C}$ in $\mathcal{D}$.** For example: if $\mathcal{C}$ is the PROP for monoids, $\hom(\mathcal{C}, (\mathrm{Set}, \times))$ is the category of monoids, while $\hom(\mathcal{C}, (\mathrm{Vect}, \otimes))$ is the category of associative algebras.

Given symmetric monoidal categories $\mathcal{C}$ and $\mathcal{D}$, $\hom(\mathcal{C}, \mathcal{D})$ is always a *symmetric monoidal category* in a natural way! E.g. we can define the tensor product of monoids, and also of associative algebras.

Even better, given symmetric monoidal categories $\mathcal{C}$ and $\mathcal{D}$, there's a symmetric monoidal category $\mathcal{C} \otimes \mathcal{D}$ such that:

$$\hom(\mathcal{C} \otimes \mathcal{D}, \mathcal{E}) \simeq \hom(\mathcal{C}, \hom(\mathcal{D}, \mathcal{E}))$$

For example, if $\mathcal{C}$ is the PROP for monoids, $\mathcal{C}^{\mathrm{op}}$ is that for comonoids, and $\mathcal{C} \otimes \mathcal{C}^{\mathrm{op}}$ is that for 'bimonoids'. A **bimonoid** is both a monoid and a comonoid, with all the comonoid operations being monoid homomorphisms (and vice versa).

Now, back to our pseudo-monad

$$\mathrm{SymmMonCat} \xrightarrow{M} \mathrm{SymmMonCat}$$

which takes any symmetric monoidal category and endows it with finite products. In a category with finite products, every object is automatically a cocommutative comonoid, thanks to

$$\Delta_x \colon x \to x \times x$$

and

$$e_x \colon x \to 1$$

And indeed....

Theorem:

$$M\mathcal{C} \simeq \mathcal{C} \otimes (\mathrm{FinSet}^{\mathrm{op}}, +)$$

where $(\mathrm{FinSet}^{\mathrm{op}}, +)$ is the PROP for cocommutative comonoids.

Philosophical Postlude

Why do algebraic theories so nicely describe a structured object in a classical world, and PROPs a structured object in a quantum world?

In our world, *physical objects* act approximately like point particles. As time passes, they trace out 1d paths in 4d spacetime: precisely the string diagrams we use to reason in symmetric monoidal categories! So, it's not surprising that symmetric monoidal categories seem like a comfortable context for *mathematical* objects.

But why are 1d paths in 4d spacetime related to symmetric monoidal categories? The full explanation involves n-categories....

A symmetric monoidal category is the same as a 4-category with one j-morphism for $j < 3$:

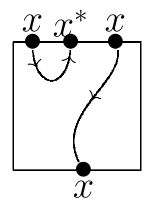
THE PERIODIC TABLE

	$n = \mathbf{0}$	$n = \mathbf{1}$	$n = \mathbf{2}$
$k = \mathbf{0}$	**sets**	**categories**	**2-categories**
$k = \mathbf{1}$	**monoids**	**monoidal categories**	**monoidal 2-categories**
$k = \mathbf{2}$	**commutative monoids**	**braided monoidal categories**	**braided monoidal 2-categories**
$k = \mathbf{3}$	‘’	**symmetric monoidal categories**	**sylleptic monoidal 2-categories**
$k = \mathbf{4}$	‘’	‘’	**symmetric monoidal 2-categories**
$k = \mathbf{5}$	‘’	‘’	‘’
$k = \mathbf{6}$	‘’	‘’	‘’

k-tuply monoidal n-categories:

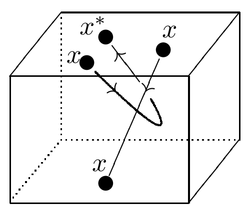
$(n + k)$-categories with only one j-morphism for $j < k$

1d tangles in codimension 1 are the morphisms of a monoidal category $1\mathrm{Tang}_1$:



This is the *free monoidal category with duals on one object.*

1d tangles in codimension 2 form a braided monoidal category $1\mathrm{Tang}_2$:

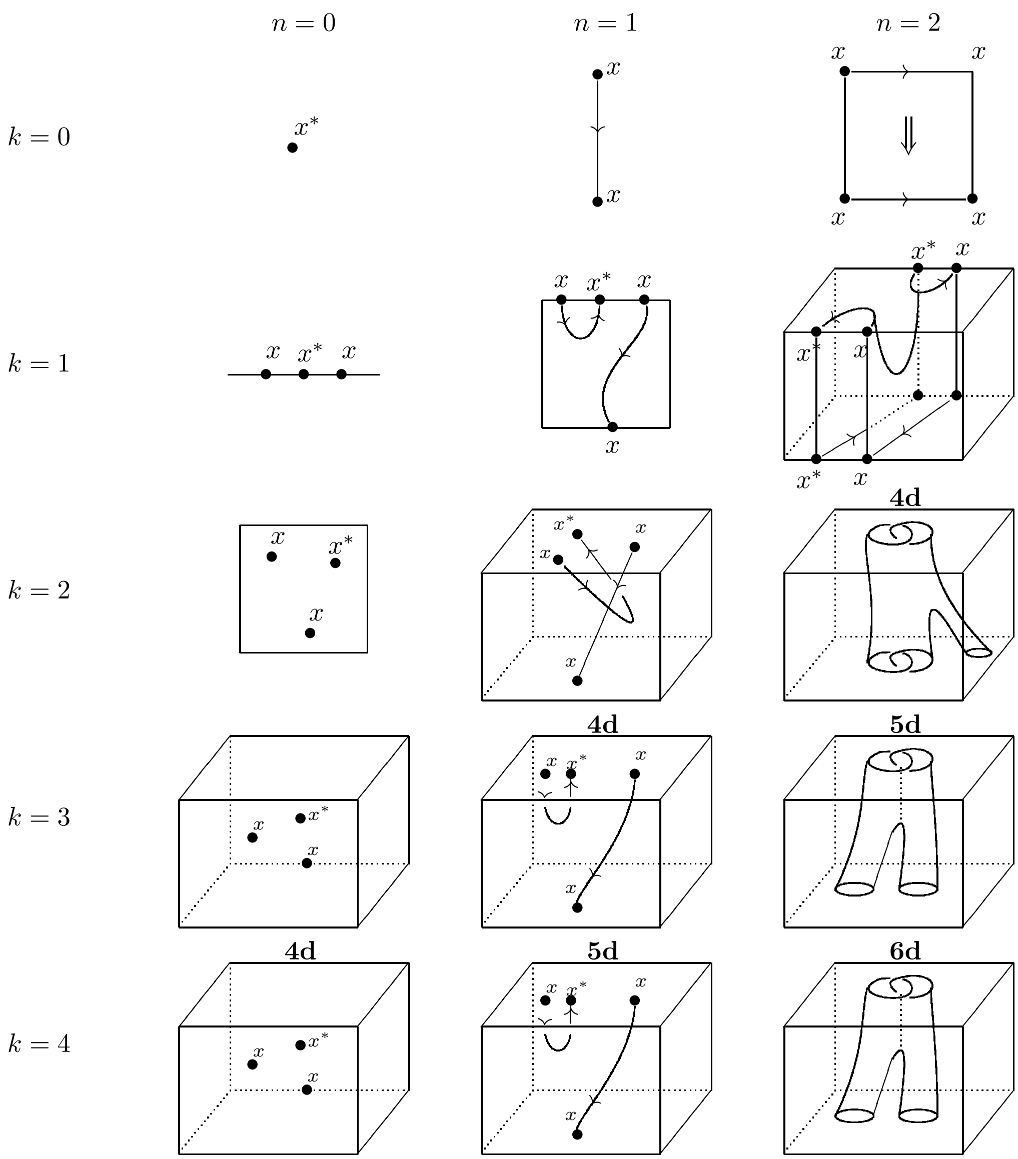


This is the *free braided monoidal category with duals on one object.*

1d tangles in codimension 3 form $1\mathrm{Tang}_3$, the *free symmetric monoidal category with duals on one object.*

In general we expect $n\mathrm{Tang}_k$ to be the free k-tuply monoidal n-category with duals on one object:

THE TANGLE HYPOTHESIS



In short: since we perceive a universe of codimension 3 objects in dimension 4, our 'universal algebra' uses 3-tuply monoidal 4-categories.

An extra simplification in classical — i.e., non-quantum — logic is that we can *duplicate and delete information.* This is not really true in the physical world, but our copying machines and wastebaskets do a pretty good job of creating this impression, so mathematicians like symmetric monoidal categories where this holds. These are *categories with finite products*, the archetypal example being Set.

But, to get closer to reality we should climb the n-categorical ladder, and learn to love the quantum universe.

References

For more details, try these online references:

John Baez, Quantum quandaries: a category-theoretic perspective,
http://math.ucr.edu/home/baez/quantum/

John Baez and James Dolan, Higher-dimensional algebra and topological quantum field theory, q-alg/9503002

John Baez and James Dolan, Categorification, math.QA/9802029

Michael Barr and Charles Wells, *Toposes, Triples and Theories*,
http://www.cwru.edu/artsci/math/wells/pub/ttt.html

Stanley Burris and H.P. Sankappanavar, *A Course in Universal Algebra*,
http://www.math.uwaterloo.ca/ snburris/htdocs/ualg.html

Aaron Lauda, Frobenius algebras and ambidextrous adjunctions, math.CT/0502550

Aaron Lauda and Hendryk Pfeiffer, Open-closed strings: two-dimensional extended TQFTs and Frobenius algebras, math.AT/0510664

F. William Lawvere, *Functorial Semantics of Algebraic Theories*,
http://www.tac.mta.ca/tac/reprints/articles/5/tr5abs.html

Robert Rosebrugh, Nicoletta Sabadini and Robert Walters, Generic commutative separable algebras and cospans of graphs,
http://www.tac.mta.ca/tac/volumes/15/6/15-06abs.html

Todd Trimble, Bar constructions,
http://math.ucr.edu/home/baez/universal/bar.html