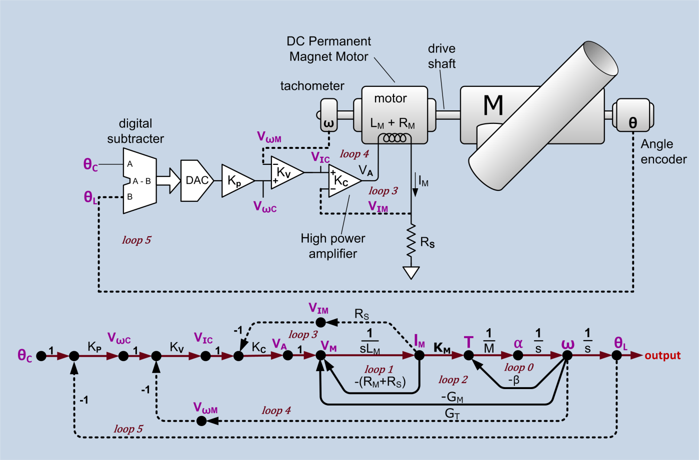
CATEGORIES IN CONTROL

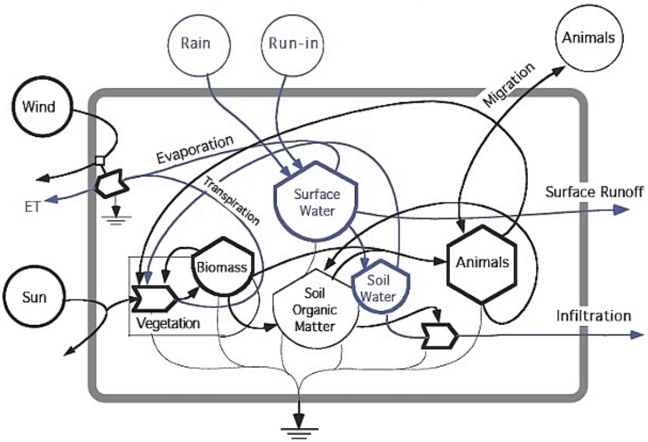


John Baez

Canadian Mathematical Society Winter Meeting

5 December 2015

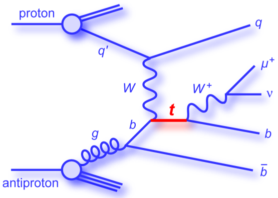
To understand ecosystems, ultimately will be to understand networks. — B. C. Patten and M. Witkamp



We need a good mathematical theory of networks.

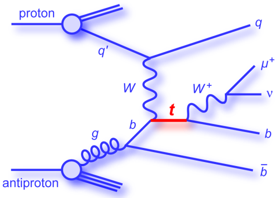
The category with vector spaces as objects and linear maps as morphisms becomes symmetric monoidal with the usual $\otimes$.

In quantum field theory, Feynman diagrams are pictures of morphisms in this symmetric monoidal category:



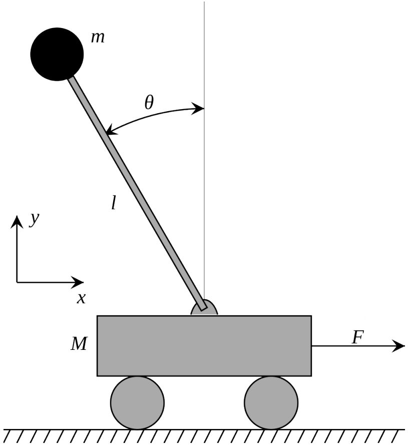
The category with vector spaces as objects and linear maps as morphisms becomes symmetric monoidal with the usual $\otimes$.

In quantum field theory, Feynman diagrams are pictures of morphisms in this symmetric monoidal category:

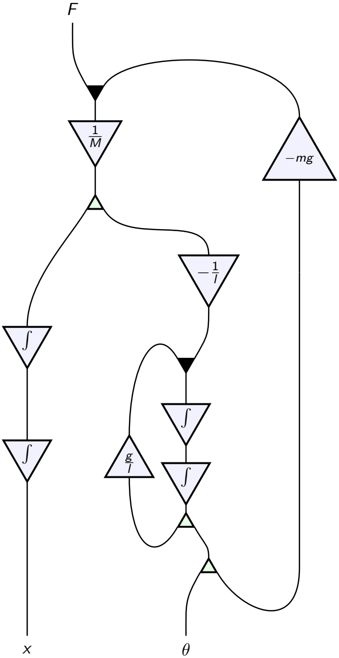


But the category of vector spaces also becomes symmetric monoidal using $\oplus$. This is important in **control theory**: the art of getting systems to do what you want. Control theorists use 'signal-flow diagrams' to describe morphisms in this symmetric monoidal category.

For example, an upside-down pendulum on a cart:



has the following signal-flow diagram...

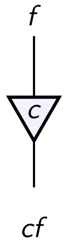


F
$\frac{1}{M}$
$-mg$
$-\frac{1}{l}$
$\int$
$\int$
$\int$
$\frac{g}{l}$
$\int$
x
θ

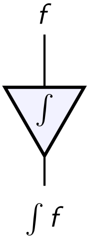
To formalize this, think of a signal as a smooth real-valued function of time:

$$f : \mathbb{R} \to \mathbb{R}$$

We can multiply a signal by a constant and get a new signal:



We can also integrate a signal:



Electrical engineers use Laplace transforms to write signals as linear combinations of exponentials:

$$f(t) = e^{-st} \quad \text{for some } s > 0$$

Then they define

$$(\textstyle\int f)(t) = \frac{e^{-st}}{s}$$

Electrical engineers use Laplace transforms to write signals as linear combinations of exponentials:

$$f(t) = e^{-st} \quad \text{for some } s > 0$$

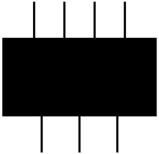
Then they define

$$(\textstyle\int f)(t) = \frac{e^{-st}}{s}$$

This lets us think of integration as a special case of scalar multiplication!

So, signal-flow diagrams are a tool for linear algebra over $k = \mathbb{R}(s)$, the field of rational functions in one real variable s.

Let us work over any commutative rig k. We start by using signal-flow diagrams with m inputs and n outputs:



to describe k-linear maps

$$F\colon k^m \to k^n$$

These signal flow diagrams are pictures of morphisms in $\mathbf{FinVect}_k$, the strict symmetric monoidal category with:

- one object k^n for each $n \in \mathbb{N}$
- k-linear maps $F\colon k^m \to k^n$ as morphisms

and with tensor product given by direct sum.

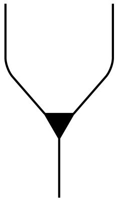
"$\mathbf{FinVect}_k$" is abuse of notation: we're talking about finitely generated free k-modules, which are finite-dimensional vector spaces when k is a field.

$\mathbf{FinVect}_k$ is generated as a symmetric monoidal category by one object, k, and 5 kinds of morphisms:

1. **Scalar multiplication** by $c \in k$

$$\begin{aligned} c\colon k &\to k \\ x &\mapsto cx \end{aligned}$$

2. **Addition**:

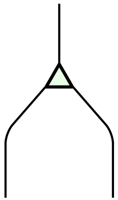


$$+: \quad \begin{array}{ccc} k^2 & \to & k \\ (x, y) & \mapsto & x + y \end{array}$$

3. **Zero**:

$$0\colon \quad \begin{array}{ccc} \{0\} & \to & k \\ 0 & \mapsto & 0 \end{array}$$

4. **Duplication**:



$$\Delta: \quad k \to k^2$$
$$x \mapsto (x, x)$$

5. **Deletion**:

$$
\begin{array}{rccc}
!: & k & \rightarrow & \{0\} \\
 & x & \mapsto & 0
\end{array}
$$

We know all the relations these generating morphisms obey:

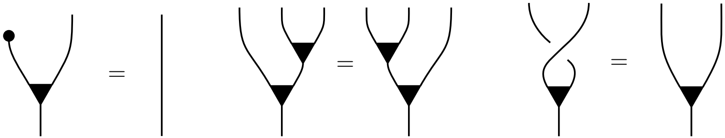
Theorem (Baez–Erbele,Wadsley–Woods)

$\mathbf{FinVect}_k$ *is the PROP for bicommutative bimonoids over* k*.*

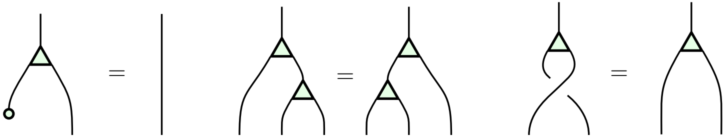
This a terse way to list relations, and to say that these imply *all* the relations.

In detail...

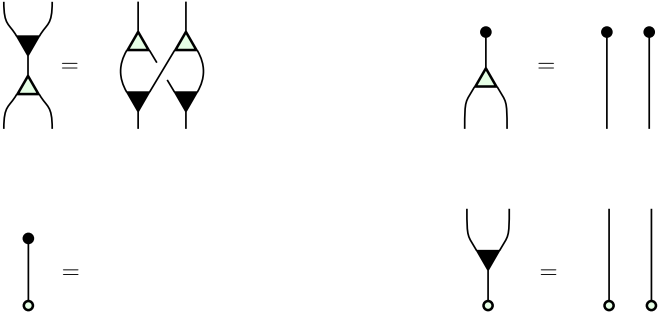
(1)–(3) Addition and zero make *k* into a commutative monoid:



(4)–(6) Duplication and deletion make *k* into a cocommutative comonoid:



(7)–(10) The monoid and comonoid structures on k fit together to form a bicommutative bimonoid:



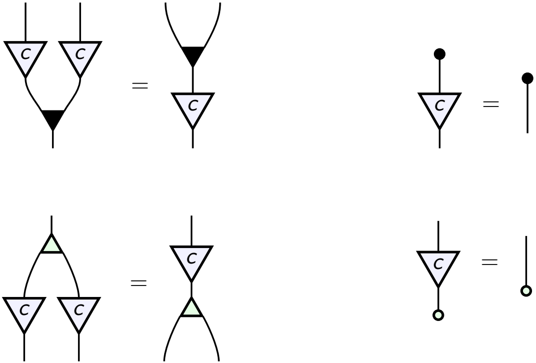
What is a bicommutative bimonoid "over k"?

For any bicommutative bimonoid A in a symmetric monoidal category, the bimonoid endomorphisms $f\colon A \to A$ can be added and composed, giving a rig $\mathrm{End}(A)$.

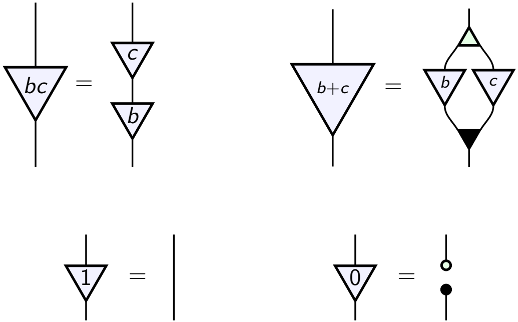
A bicommutative bimonoid **over *k*** is one equipped with a rig homomorphism

$$\Phi\colon k \to \mathrm{End}(A)$$

(11)–(14) Saying that Φ sends each $c \in k$ to a bimonoid homomorphism means that these extra relations hold:

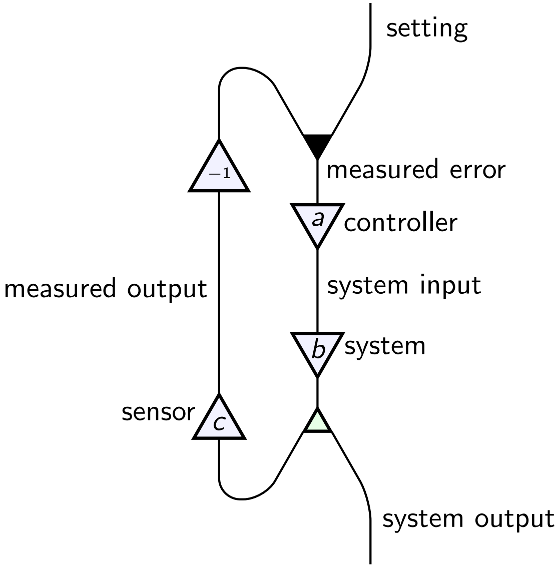


(15)–(18) Saying that Φ is a rig homomorphism means that these extra relations hold:



So, these are all the relations in $\mathbf{FinVect}_k$.

But control theory also needs more general signal-flow diagrams with ‘feedback loops’:



Feedback is the most important concept in control theory: letting the output of a system affect its input. For this we should let wires 'bend back':

These aren't linear maps — they're linear *relations!*

A **linear relation** $F\colon U \rightsquigarrow V$ from a vector space U to a vector space V is a linear subspace $F \subseteq U \oplus V$.

A **linear relation** $F\colon U \rightsquigarrow V$ from a vector space U to a vector space V is a linear subspace $F \subseteq U \oplus V$.

We can compose linear relations in the usual way we compose relations. There is a symmetric monoidal category $\mathbf{FinRel}_k$ with:

- one object k^n for each $n \in \mathbb{N}$
- linear relations $F\colon k^m \rightsquigarrow k^n$ as morphisms

and with tensor product given by direct sum. It has $\mathbf{FinVect}_k$ as a symmetric monoidal subcategory.

A **linear relation** $F\colon U \rightsquigarrow V$ from a vector space U to a vector space V is a linear subspace $F \subseteq U \oplus V$.

We can compose linear relations in the usual way we compose relations. There is a symmetric monoidal category $\mathbf{FinRel}_k$ with:

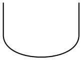
- one object k^n for each $n \in \mathbb{N}$
- linear relations $F\colon k^m \rightsquigarrow k^n$ as morphisms

and with tensor product given by direct sum. It has $\mathbf{FinVect}_k$ as a symmetric monoidal subcategory.

Fully general signal-flow diagrams are pictures of morphisms in $\mathbf{FinRel}_k$, typically with $k = \mathbb{R}(s)$.

Besides the generators of $\mathbf{FinVect}_k$ we only need two more morphisms to generate $\mathbf{FinRel}_k$:

6. The **cup**:



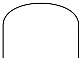
This is the linear relation

$$\cup \colon k^2 \rightsquigarrow \{0\}$$

given by

$$\cup \ = \ \{(x,x,0) : x \in k\} \ \subseteq \ k^2 \oplus \{0\}$$

7. The **cap**:



This is the linear relation

$$\cap\colon \{0\} \rightsquigarrow k^2$$

given by

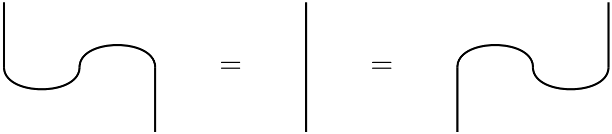
$$\cap \;=\; \{(0,x,x) : x \in k\} \;\subseteq\; \{0\} \oplus k^2$$

Theorem (Baez–Erbele, Bonchi–Sobociński–Zanasi)

FinRel$_k$ *is the free symmetric monoidal category on a pair of interacting bimonoids over* k.

Besides the relations we've seen so far, this statement summarizes the following extra relations:

(19)–(20) $\cap$ and $\cup$ obey the zigzag relations:



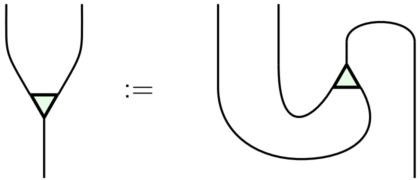
It follows that $(\mathbf{FinRel}_k, \oplus)$ becomes a dagger-compact category, so we can 'turn around' any morphism $F\colon U \rightsquigarrow V$ and get its **adjoint** $F^\dagger\colon V \rightsquigarrow U$:

$$F^\dagger = \{(v, u) : (u, v) \in F\}$$

It follows that $(\mathbf{FinRel}_k, \oplus)$ becomes a dagger-compact category, so we can 'turn around' any morphism $F \colon U \rightsquigarrow V$ and get its **adjoint** $F^\dagger \colon V \rightsquigarrow U$:

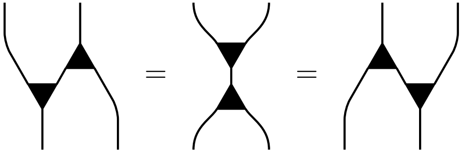
$$F^\dagger = \{(v, u) : (u, v) \in F\}$$

For example, turning around duplication $\Delta \colon k \to k \oplus k$ gives **coduplication**, $\Delta^\dagger \colon k \oplus k \rightsquigarrow k$:

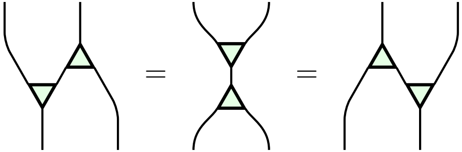


$$\Delta^\dagger = \{(x, x, x)\} \subseteq k^2 \oplus k$$

(21)–(22) $(k, +, 0, +^\dagger, 0^\dagger)$ is a Frobenius monoid:



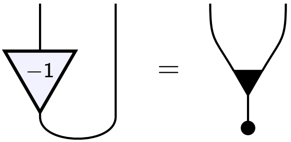
(23)–(24) $(k, \Delta^\dagger, !^\dagger, \Delta, !)$ is a Frobenius monoid:



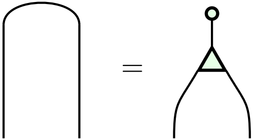
(25)–(26) The Frobenius monoid $(k, +, 0, +^{\dagger}, 0^{\dagger})$ is extra-special:

(27)–(28) The Frobenius monoid $(k, \Delta^{\dagger}, !^{\dagger}, \Delta, !)$ is extra-special:

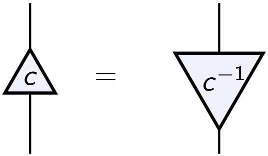
(29) $\cup$ with a factor of -1 inserted can be expressed in terms of $+$ and 0:



(30) $\cap$ can be expressed in terms of Δ and !:



(31) For any $c \in k$ with $c \neq 0$, scalar multiplication by c^{-1} is the adjoint of scalar multiplication by c:



This is part of a larger story:

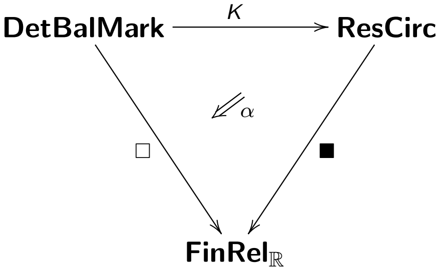
An electrical circuit made of resistors, resistors and capacitors gives a linear relation between its input and output voltages and currents. Baez and Fong showed this gives a symmetric monoidal functor:

$$\mathbf{Circ} \xrightarrow{\ \blacksquare\ } \mathbf{FinRel}_{\mathbb{R}(s)}$$

This gives the 'semantics' for circuit diagrams. For circuits made only of resistors, we have

$$\mathbf{ResCirc} \xrightarrow{\ \blacksquare\ } \mathbf{FinRel}_{\mathbb{R}}$$

Similarly, Baez, Fong and Pollard showed that the steady states of an open detailed balanced Markov process determine a linear relation between its input and output populations and flows. This gives a symmetric monoidal functor $\square\colon \mathbf{DetBalMark} \to \mathbf{FinRel}_{\mathbb{R}}$ fitting into this diagram:



In short, we can reduce the ‘steady-state semantics’ of detailed balanced Markov processes to that of circuits made of resistors!