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:

![Feynman diagram](image)

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...
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

\[ (\int f)(t) = \frac{e^{-st}}{s} \]
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

\[ (\int f)(t) = \frac{e^{-st}}{s} \]

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:

![Signal-flow diagram]

to describe $k$-linear maps

$$F : k^m \rightarrow k^n$$
These signal flow diagrams are pictures of morphisms in $\text{FinVect}_k$, the strict symmetric monoidal category with:

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

and with tensor product given by direct sum.

“$\text{FinVect}_k$” is abuse of notation: we’re talking about finitely generated free $k$-modules, which are finite-dimensional vector spaces when $k$ is a field.
**FinVect**}_k is generated as a symmetric monoidal category by one object, \( k \), and 5 kinds of morphisms:

1. **Scalar multiplication** by \( c \in k \)

\[
c : k \rightarrow k
\]
\[
x \mapsto cx
\]
2. **Addition:**

\[
+ : \quad k^2 \quad \rightarrow \quad k \\
(x, y) \quad \mapsto \quad x + y
\]
3. Zero:

\[ 0: \{0\} \rightarrow k \]

\[ 0 \Leftrightarrow 0 \]
4. Duplication:

\[\Delta: \quad k \rightarrow k^2\]
\[x \leftrightarrow (x, x)\]
5. Deletion:

\[!: k \rightarrow \{0\}\]

\[x \mapsto 0\]
We know all the relations these generating morphisms obey:

**Theorem (Baez–Erbele, Wadsley–Woods)**

Finite-dimensional $k$-modules 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:
The monoid and comonoid structures on $k$ fit together to form a bicommutative bimonoid:

\[(7)\text{–}(10)\]

\[\begin{align*}
\begin{array}{c}
\quad\
\end{array}
\end{align*} = \begin{array}{c}
\quad\
\end{array} \quad \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array}
\]

\[\begin{align*}
\begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array}
\]

\[\begin{align*}
\begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array}
\]

\[\begin{align*}
\begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array} = \begin{array}{c}
\quad\
\end{array}
\]
What is a bicommutative bimonoid “over $k$”?

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

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

$$\Phi : k \to \text{End}(A)$$
(11)–(14) Saying that $\Phi$ sends each $c \in k$ to a bimonoid homomorphism means that these extra relations hold:
(15)–(18) Saying that $\Phi$ is a rig homomorphism means that these extra relations hold:

\[
\begin{align*}
bc &= c \quad & b+c &= b \quad & 1 &= 0 \\
\end{align*}
\]

So, these are all the relations in $\text{FinVect}_k$. 
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 $F: U \leadsto V$ from a vector space $U$ to a vector space $V$ is a linear subspace $F \subseteq U \oplus V$. 
A **linear relation** \( F : U \leadsto V \) 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 \( \text{FinRel}_k \) with:

- one object \( k^n \) for each \( n \in \mathbb{N} \)
- linear relations \( F : k^m \leadsto k^n \) as morphisms

and with tensor product given by direct sum. It has \( \text{FinVect}_k \) as a symmetric monoidal subcategory.
A **linear relation** $F : U \rightsquigarrow V$ 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 $\text{FinRel}_k$ with:

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

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

Fully general signal-flow diagrams are pictures of morphisms in $\text{FinRel}_k$, typically with $k = \mathbb{R}(s)$. 
Besides the generators of $\text{FinVect}_k$ we only need two more morphisms to generate $\text{FinRel}_k$:

6. The cup:

$$\cup$$

This is the linear relation

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

given by

$$\cup = \{(x, x, 0) : x \in k\} \subseteq k^2 \oplus \{0\}$$
7. The **cap**:

\[ \cap \]

This is the linear relation

\[ \cap : \{0\} \leadsto k^2 \]

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 \((\text{FinRel}_k, \oplus)\) becomes a dagger-compact category, so we can ‘turn around’ any morphism \(F: U \rightsquigarrow V\) and get its adjoint \(F^\dagger: V \rightsquigarrow U:\)

\[
F^\dagger = \{(v, u) : (u, v) \in F\}
\]
It follows that \((\text{FinRel}_k, \oplus)\) becomes a dagger-compact category, so we can ‘turn around’ any morphism \(F : U \rightsquigarrow V\) and get its adjoint \(F^\dagger : V \rightsquigarrow U\):

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

For example, turning around duplication \(\Delta : k \to k \oplus k\) gives coduplication, \(\Delta^\dagger : k \oplus k \rightsquigarrow k\):

\[
\Delta^\dagger = \{ (x, x, x) \} \subseteq k^2 \oplus k
\]
(21)-(22) \((k, +, 0, +^\dagger, 0^\dagger)\) is a Frobenius monoid:

\[
\begin{array}{ccc}
\quad & \quad & \quad \\
\quad & \quad & \quad \\
\end{array}
\]

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

\[
\begin{array}{ccc}
\quad & \quad & \quad \\
\quad & \quad & \quad \\
\end{array}
\]
(25)–(26) The Frobenius monoid \((k, +, 0, +^\dagger, 0^\dagger)\) is extra-special:
\[
\begin{array}{c}
\begin{array}{c}
\text{\includegraphics[width=1cm]{monoid1}}
\end{array}
\quad = \quad \\
\begin{array}{c}
\text{\includegraphics[width=1cm]{monoid2}}
\end{array}
\end{array}
\]

(27)–(28) The Frobenius monoid \((k, \Delta^\dagger, !^\dagger, \Delta, !)\) is extra-special:
\[
\begin{array}{c}
\begin{array}{c}
\text{\includegraphics[width=1cm]{monoid3}}
\end{array}
\quad = \quad \\
\begin{array}{c}
\text{\includegraphics[width=1cm]{monoid4}}
\end{array}
\end{array}
\]
(29) $\cup$ with a factor of $-1$ inserted can be expressed in terms of $+$ and $0$:

\[
\begin{align*}
\begin{array}{c}
\text{\includegraphics{image1}}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{c}
\text{\includegraphics{image2}}
\end{array}
\end{align*}
\]

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

\[
\begin{align*}
\begin{array}{c}
\text{\includegraphics{image3}}
\end{array}
\end{align*}
\]

\[
\begin{align*}
\begin{array}{c}
\text{\includegraphics{image4}}
\end{array}
\end{align*}
\]
(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:

\[
\text{Circ} \longrightarrow \text{FinRel}_\mathbb{R}(s)
\]

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

\[
\text{ResCirc} \longrightarrow \text{FinRel}_\mathbb{R}
\]
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 $\boxtimes : \text{DetBalMark} \to \text{FinRel}_\mathbb{R}$ 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!