STRUCTURED COSPANS AND DOUBLE CATEGORIES

John Baez, Kenny Courser, Christina Vasilakopoulou
1 April 2020
Throughout science and engineering, people use *networks*, drawn as boxes connected by wires:

So, they’re using categories! Which categories are these?
Networks of a particular kind, with specified inputs and outputs, can be seen as morphisms in a particular symmetric monoidal category:

\[ X \xrightarrow{\text{stuff can flow in or out}} \cdot \xleftarrow{\text{we can combine systems to form larger systems by composition and tensoring}} \cdot Y \]
Networks of a particular kind, with specified inputs and outputs, can be seen as morphisms in a particular symmetric monoidal category:

\[ X \rightarrow \rightarrow Y \]

Such networks let us describe “open systems”, meaning systems where:

- stuff can flow in or out;
- we can combine systems to form larger systems by composition and tensoring.
We can describe networks with inputs and outputs using cospans with extra structure. For example, this:

\[
\begin{array}{c}
X \quad \bullet \\
\downarrow \quad \downarrow \\
S \\
\uparrow \quad \uparrow \\
Y
\end{array}
\]

is really a cospan of finite sets:

\[
\begin{array}{c}
S \\
\downarrow \quad \downarrow \\
X \quad \bullet \\
\downarrow \quad \downarrow \\
Y
\end{array}
\]

where \( S \) is decorated with extra structure: edges making \( S \) into the vertices of a graph.
Fong invented ‘decorated cospans’ to make this precise:

Fong invented ‘decorated cospans’ to make this precise:


We’ve used them to study many kinds of networks.
Electrical circuits:

Markov processes:

Petri nets with rates:

Now Kenny Courser has developed a simpler formalism: ‘structured cospans’.

We have redone most of the previous work using structured cospans:


Let’s see how structured cospans work in an example: Petri nets with rates.
A **Petri net with rates** is a diagram like this:

\[(0, \infty) \xleftarrow{r} T \xrightarrow{s} \mathbb{N}[S]\]

where \(S\) and \(T\) are finite sets, and \(\mathbb{N}[S]\) is the set of finite formal sums of elements of \(S\).
A **Petri net with rates** is a diagram like this:

$$(0, \infty) \xleftarrow{r} T \xrightarrow{s} \mathbb{N}[S]$$

where $S$ and $T$ are finite sets, and $\mathbb{N}[S]$ is the set of finite formal sums of elements of $S$.

We call elements of $S$ **places**, elements of $T$ **transitions**, and $r(t)$ the **rate constant** of the transition $t \in T$. 
Given a Petri net with rates, we can write down a rate equation describing dynamics. Example: the SIR model of infectious disease:

\[
\begin{align*}
\frac{dS}{dt} &= -r_1 SI \\
\frac{dI}{dt} &= r_1 SI - r_2 I \\
\frac{dR}{dt} &= r_2 I
\end{align*}
\]
Given a Petri net with rates, we can write down a rate equation describing dynamics. Example: the SIR model of infectious disease:

\[
\begin{align*}
\frac{dS}{dt} &= -r_1 SI \\
\frac{dl}{dt} &= r_1 SI - r_2 l \\
\frac{dR}{dt} &= r_2 l
\end{align*}
\]
We can also define an *open* Petri net with rates:

\[
\begin{align*}
\frac{dS}{dt} &= -r_1 SI + i_1 + i_2 \\
\frac{dI}{dt} &= r_1 SI - r_2 I + i_3 \\
\frac{dR}{dt} &= r_2 I - o_1
\end{align*}
\]
We can also define an *open* Petri net with rates:

\[ \begin{align*}
\frac{dS}{dt} &= -r_1 Sl + i_1 + i_2 \\
\frac{dl}{dt} &= r_1 Sl - r_2 l + i_3 \\
\frac{dR}{dt} &= r_2 l - o_1
\end{align*} \]
There is a category Open(Petri$_r$) where objects are finite sets and morphisms are open Petri nets.

There is a functor

\[
\mathcal{F} : \text{Open(Petri}_r) \to \text{Dynam}
\]

sending each open Petri net to its rate equation, which is treated as a morphism in a category Dynam. Since

\[
\mathcal{F}(PQ) = \mathcal{F}(P) \circ \mathcal{F}(Q)
\]

the process of extracting the rate equation from an open Petri net is ‘compositional’.
How do we build the category $\text{Open}(\text{Petri}_r)$, with open Petri nets as morphisms?

Using the theory of structured cospans!
Given a functor

\[ L : A \to X \]

a **structured cospan** is a diagram

\[
\begin{array}{c}
X \\
\downarrow^i \\
L(a) \\
\uparrow_o \\
\end{array} \quad \begin{array}{c}
\downarrow^o \\
L(b) \\
\end{array}
\]

Think of \( A \) as a category of objects with 'less structure', and \( X \) as a category of objects with 'more structure'. \( L \) is often a left adjoint.
Given a functor

\[ L : A \to X \]

a **structured cospan** is a diagram

\[ \begin{array}{c}
X \\
\left\arrow{i} \quad \left\arrow{o} \\
L(a) \quad L(b)
\end{array} \]

Think of A as a category of objects with ‘less structure’, and X as a category of objects with ‘more structure’. L is often a left adjoint.
Theorem (Kenny Courser, JB)

Let $A$ and $X$ be categories with finite colimits, and $L: A \to X$ a left adjoint.

Then there is a symmetric monoidal category $\mathcal{L}Csp(X)$ where:

- an object is an object of $A$
- a morphism is an isomorphism class of structured cospans:

$$
\begin{array}{c}
X \\
\downarrow i \quad \downarrow o \\
L(a) & & L(b)
\end{array}
$$
Here two structured cospans are **isomorphic** if there is a commuting diagram of this form:

\[
\begin{array}{c}
\text{\(X\)} \\
\text{\(\downarrow\)} \\
\text{\(L(a) \simeq f \ L(b)\)} \\
\text{\(X'\)} \\
\end{array}
\]

\[
\begin{array}{cc}
\text{\(\downarrow\)} & \text{\(\downarrow\)} \\
\text{\(i\)} & \text{\(o\)} \\
\text{\(L(a) \simeq f \ L(b)\)} \\
\text{\(i'\)} & \text{\(o'\)} \\
\end{array}
\]
Given two structured cospans

\[
\begin{align*}
&x \\
&\quad \downarrow i \\
&\quad L(a_1) \\
&\quad \downarrow o \\
&\quad L(a_2) \\
\end{align*}
\begin{align*}
&y \\
&\quad \downarrow i' \\
&\quad L(a_2) \\
&\quad \downarrow o' \\
&\quad L(a_3) \\
\end{align*}
\]

we compose them by taking a pushout in the category \( X \):

\[
\begin{align*}
&x +_{L(a_2)} y \\
&\quad \downarrow \\
&\quad L(a_1) \\
&\quad i \\
&\quad o \\
\end{align*}
\begin{align*}
&\quad \downarrow i' \\
&\quad L(a_2) \\
&\quad \downarrow o' \\
&\quad L(a_3) \\
\end{align*}
\]
To tensor structured cospans:

\[
\begin{array}{c}
\xymatrix{
X \ar[dr]^-o \ar[dl]^-i \\
L(a) & L(b) \\
}
\end{array}
\quad
\begin{array}{c}
\xymatrix{
X' \ar[dr]^-{o'} \ar[dl]^-{i'} \\
L(a') & L(b') \\
}
\end{array}
\]

we use coproducts in \(A\) and \(X\):

\[
\begin{array}{c}
\xymatrix{
X + X' \ar[dr]^-{o + o'} \ar[dl]^-{i + i'} \\
L(a) + L(a') & L(b) + L(b') \\
}
\end{array}
\]

and the fact that \(L: A \to X\) preserves coproducts.
This theorem applies to many examples, giving structured cospan categories whose morphisms are:

- open electrical circuits
- open Markov processes
- open Petri nets
- open Petri nets with rates

etcetera.
This theorem applies to many examples, giving structured cospan categories whose morphisms are:

- open electrical circuits
- open Markov processes
- open Petri nets
- open Petri nets with rates

etcetera.

In all these examples \( A \) and \( X \) have finite colimits and \( L: A \to X \) is a left adjoint, so the theorem applies.
This theorem applies to many examples, giving structured cospan categories whose morphisms are:

- open electrical circuits
- open Markov processes
- open Petri nets
- open Petri nets with rates

etcetera.

In all these examples $A$ and $X$ have finite colimits and $L: A \rightarrow X$ is a left adjoint, so the theorem applies.

Let’s see what it looks like for open Petri nets with rates.
There is a category Petri\(_r\) where objects are Petri nets with rates, and morphisms are diagrams like this:

\[
\begin{array}{ccc}
(0, \infty) & \xrightarrow{\quad r \quad} & T & \overset{s}{\longrightarrow} & \mathbb{N}[S] \\
& r' & f \downarrow & & \downarrow \mathbb{N}[g] \\
& & T' & \overset{s'}{\longrightarrow} & \mathbb{N}[S'] \quad t'
\end{array}
\]

where the square involving \(s\) and \(s'\) commutes, as does the square involving \(t\) and \(t'\).
There is a functor $R : \text{Petri}_r \to \text{FinSet}$ sending any Petri net with rates to its underlying set of places.

This has a left adjoint $L : \text{FinSet} \to \text{Petri}_r$ sending any set to the Petri net with that set of places, and no transitions.
There is a functor $R : \text{Petri}_r \to \text{FinSet}$ sending any Petri net with rates to its underlying set of places.

This has a left adjoint $L : \text{FinSet} \to \text{Petri}_r$ sending any set to the Petri net with that set of places, and no transitions.

In this example, a structured cospan

\[
\begin{array}{c}
\overset{i}{x} \\
\downarrow \\
L(a) \quad \overset{o}{x} \\
\downarrow \\
L(b)
\end{array}
\]

is an **open Petri net with rates**:

\[
\begin{array}{c}
\overset{a}{x} \\
\downarrow \\
1.3 \\
\downarrow \\
\overset{b}{x}
\end{array}
\]
We can compose open Petri nets with rates:

by identifying the outputs of the first with the inputs of the second:
To tensor open Petri nets with rates:

\[ a + a' \quad b + b' \]

we set them side by side:
What if we want to use actual structured cospans, rather than isomorphism classes? We must do this to point to a *specific* place or transition in an open Petri net:
What if we want to use actual structured cospans, rather than isomorphism classes? We must do this to point to a specific place or transition in an open Petri net:

![Diagram](image)

Then we should use a symmetric monoidal double category!

A double category has figures like this:

\[
\begin{array}{ccc}
A & \xrightarrow{M} & B \\
\downarrow^f & \downarrow^\alpha & \downarrow^g \\
C & \xrightarrow{N} & D
\end{array}
\]

So, it has:

- **objects** such as \( A, B, C, D \),
A double category has figures like this:

\[
\begin{array}{ccc}
A & \xrightarrow{M} & B \\
\downarrow{f} & \downarrow{\alpha} & \downarrow{g} \\
C & \xrightarrow{N} & D
\end{array}
\]

So, it has:

- **objects** such as \(A, B, C, D\),
- **vertical 1-morphisms** such as \(f\) and \(g\),
- **2-morphisms** such as \(\alpha\).
A double category has figures like this:

\[
\begin{array}{c}
\begin{array}{ccc}
A & \xrightarrow{M} & B \\
| & \downarrow f & \downarrow \alpha & \downarrow g \\
C & \xrightarrow{N} & D
\end{array}
\end{array}
\]

So, it has:

- **objects** such as \(A, B, C, D\),
- **vertical 1-morphisms** such as \(f\) and \(g\),
- **horizontal 1-cells** such as \(M\) and \(N\),
- **2-morphisms** such as \(\alpha\).
2-morphisms can be composed vertically and horizontally, and the interchange law holds:

![Diagram](image)

Vertical composition is strictly associative and unital, but horizontal composition is not.
Theorem (Kenny Courser, JB)

Let $A$ and $X$ be categories with finite colimits, and $L: A \to X$ a left adjoint.

Then there is a symmetric monoidal double category $\mathcal{L}sp(X)$ where:

- an object is an object of $A$
- a vertical 1-morphism is a morphism of $A$
- a horizontal 1-cell is a structured cospan $L(a) \xrightarrow{i} x \xleftarrow{o} L(b)$
- a 2-morphism is a commutative diagram

```
\[
\begin{array}{ccc}
L(a) & \xrightarrow{i} & x & \xleftarrow{o} & L(b) \\
\downarrow{L(f)} & & \downarrow{h} & & \downarrow{L(g)} \\
L(a') & \xrightarrow{i'} & x' & \xleftarrow{o'} & L(b')
\end{array}
\]
```
Horizontal composition is defined using pushouts in $X$; composing these:

\[
\begin{align*}
L(a) & \to x & L(b) & \leftarrow L(c) \\
\downarrow & & \downarrow & \\
L(a') & \to x' & L(b') & \leftarrow L(c')
\end{align*}
\]

gives this:

\[
\begin{align*}
L(a) & \to x +_{L(b)} y & L(c) & \leftarrow L(c') \\
\downarrow & & \downarrow & \\
L(a') & \to x' +_{L(b')} y' & L(c')
\end{align*}
\]

Vertical composition is straightforward.
Tensoring uses binary coproducts in both \(A\) and \(X\), and the fact that \(L : A \rightarrow X\) preserves these:

\[
\begin{align*}
L(a_1) &\rightarrow x_1 \leftarrow L(b_1) & L(a'_1) &\rightarrow x'_1 \leftarrow L(b'_1) \\
\downarrow & & \downarrow \otimes & & \downarrow \\
L(a_2) &\rightarrow x_2 \leftarrow L(b_2) & L(a'_2) &\rightarrow x'_2 \leftarrow L(b'_2) \\
\end{align*}
\]

\[
L(a_1 + a'_1) \rightarrow x_1 + x'_1 \leftarrow L(b_1 + b'_1) = \\
\downarrow & & \downarrow & & \downarrow \\
L(a_2 + a'_2) \rightarrow x_2 + x'_2 \leftarrow L(b_2 + b'_2)
\]
In the case of open Petri nets, a 2-morphism can map this horizontal 1-cell:

![Original Diagram]

to this:

![Diagram after Mapping]
In the case of open Petri nets, a 2-morphism can map this horizontal 1-cell:

![Diagram 1](image1)

to this:

![Diagram 2](image2)

or this:

![Diagram 3](image3)
In summary:

- Symmetric monoidal categories are a good formalism for describing open systems — treating them as morphisms.
- Symmetric monoidal double categories are good for describing open systems precisely, not just up to isomorphism. They let us study maps between open systems.
- Structured cospans are good for building both of these things.
In summary:

- Symmetric monoidal categories are a good formalism for describing open systems — treating them as morphisms.

- Symmetric monoidal double categories are good for describing open systems precisely, not just up to isomorphism.
In summary:

- Symmetric monoidal categories are a good formalism for describing open systems — treating them as morphisms.

- Symmetric monoidal double categories are good for describing open systems precisely, not just up to isomorphism. They let us study maps between open systems.
In summary:

- Symmetric monoidal categories are a good formalism for describing open systems — treating them as morphisms.

- Symmetric monoidal double categories are good for describing open systems precisely, not just up to isomorphism. They let us study maps between open systems.

- Structured cospans are good for building both of these things.