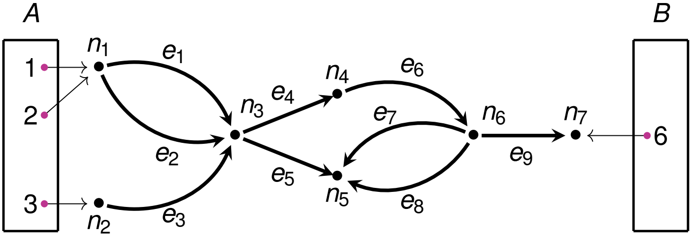
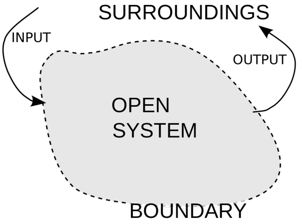
STRUCTURED vs DECORATED COSPANS



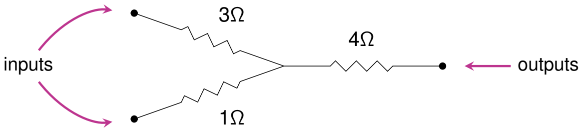
John Baez, Kenny Courser, Christina Vasilakopoulou

ACT2022

In 2010 I started thinking about open systems and networks, hoping that category theory could help us understand these.

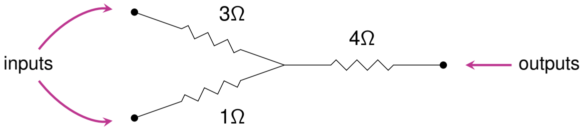


Around 2012, I asked Brendan Fong to create and study a category having “open electrical circuits” as morphisms:



He invented the theory of “decorated cospans” to do this.

Around 2012, I asked Brendan Fong to create and study a category having “open electrical circuits” as morphisms:



He invented the theory of “decorated cospans” to do this.

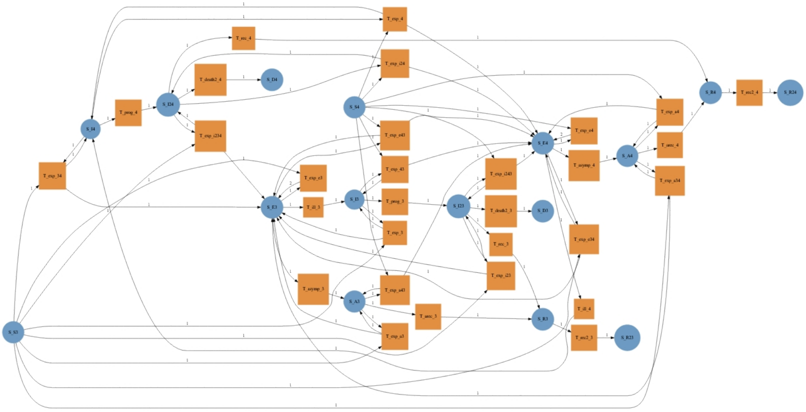
In 2018 Kenny Courser and I came up with "structured cospans", which are often simpler. Later, Christina Vasilakopolou helped us generalize the original theory of decorated cospans and relate it to structured cospans.

GSRT
GENERAL SECRETARIAT FOR RESEARCH AND TECHNOLOGY
H.F.R.I.
Hellenic Foundation for Research & Innovation

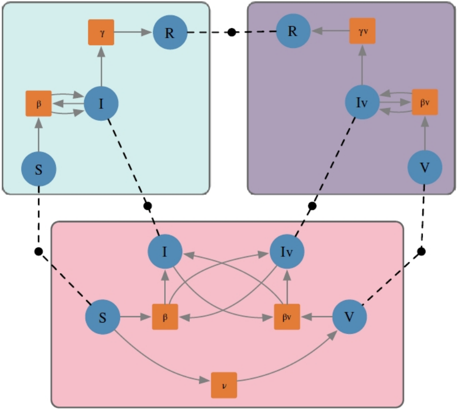
Structured and decorated cospans have by now been used to study and design many kinds of open systems:

- open graphs
- open Petri nets
- open electrical circuits
- open Markov processes
- open dynamical systems
- open Petri nets with rates ($\cong$ chemical reaction networks)
- open stock-flow diagrams

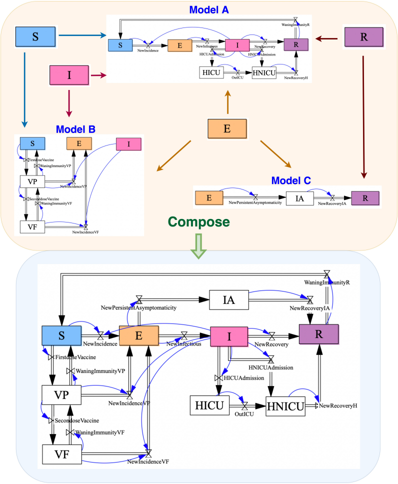
In 2020, Evan Patterson implemented structured cospans in AlgebraicJulia. Using this system to compose open Petri nets with rates, James Fairbanks and Micah Halter reconstructed some of a COVID-19 model used by the UK government.



In 2022, Sophie Libkind, Andrew Baas, Micah Halter, Evan Patterson and James Fairbanks used AlgebraicJulia to build models of epidemiology using open Petri nets with rates:

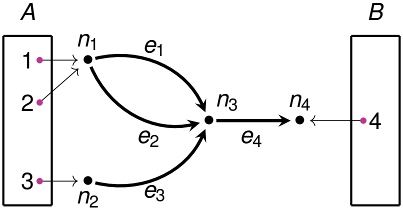


Also in 2022, Xiaoyan Li, Sophie Libkind, Nathaniel Osgood, Evan Patterson and I used AlgebraicJulia to create software for collaboratively building epidemiology models by composing open stock-flow diagrams:

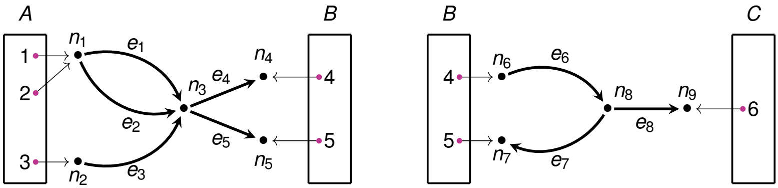


The simplest example of both structured and decorated cospans: “open graphs”.

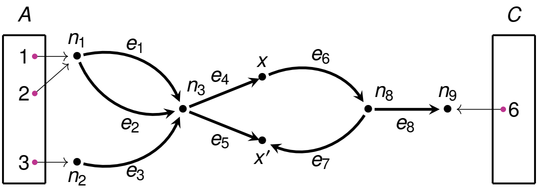
Here is an open graph with inputs A and outputs B:



We can compose open graphs by gluing them end to end:

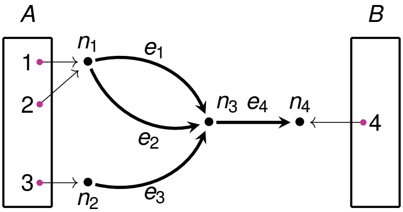


obtaining this:

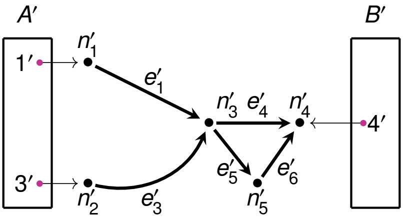


This composition is associative *up to isomorphism*. But what do we mean by 'isomorphism' here?

There is a *category* of open graphs. For example there is a morphism from this one:



to this one:



So:

- There are morphisms *between* open graphs, and composition of these morphisms is associative.
- Open graphs are *themselves* morphisms: we can compose them by gluing them end to end, and this composition is associative up to isomorphism.

The structure that captures all this is a “double category”.

A double category has figures like this:

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

So, it has:

- **objects** such as $A, B, C, D,$

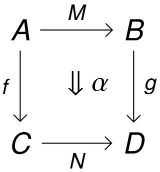
A double category has figures like this:

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

So, it has:

- **objects** such as A, B, C, D,
- **vertical 1-morphisms** such as f and g,

A double category has figures like this:



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 α.

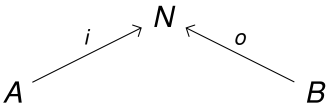
Vertical 1-morphisms can be composed. Horizontal 1-cells can be composed. 2-morphisms can be composed vertically and horizontally, and the interchange law holds:

$$
\begin{array}{ccc}
A & \xrightarrow{M} & B \\
f\downarrow & \Downarrow\alpha & \downarrow g \\
D & \xrightarrow[N]{} & E
\end{array}
\qquad
\begin{array}{ccc}
B & \xrightarrow{M'} & C \\
g\downarrow & \Downarrow\beta & \downarrow h \\
E & \xrightarrow[N']{} & F
\end{array}
$$

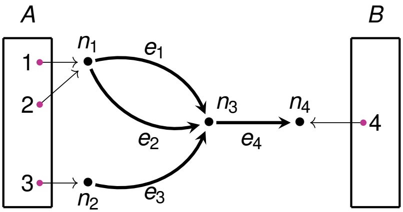
$$
\begin{array}{ccc}
D & \xrightarrow{N} & E \\
f'\downarrow & \Downarrow\alpha' & \downarrow g' \\
G & \xrightarrow[O]{} & H
\end{array}
\qquad
\begin{array}{ccc}
E & \xrightarrow{N'} & F \\
g'\downarrow & \Downarrow\beta' & \downarrow h' \\
H & \xrightarrow[P]{} & I
\end{array}
$$

Vertical composition is strictly associative and unital. Horizontal composition obeys these laws only up to isomorphism.

How can we construct the double category of open graphs?
The key is that an open graph is a **cospan** of sets:



where the **apex** N is equipped with extra structure:



Given a set N, a **graph on N** is a set E of **edges** and two functions giving the **source** and **target** of each edge:

$$E \underset{t}{\overset{s}{\rightrightarrows}} N$$

An **open graph** is a cospan of sets:

$$A \xrightarrow{\;i\;} N \xleftarrow{\;o\;} B$$

together with a graph on N.

Given a set N, a **graph on N** is a set E of **edges** and two functions giving the **source** and **target** of each edge:

$$E \underset{t}{\overset{s}{\rightrightarrows}} N$$

An **open graph** is a cospan of sets:

$$A \xrightarrow{\;i\;} N \xleftarrow{\;o\;} B$$

together with a graph on N.

But how *in general* do we equip the apex of a span with extra data?

There are at least two ways to equip an object of a category A with extra data:

- **"Structuring."** Given a right adjoint $R \colon \mathsf{X} \to \mathsf{A}$, we can give $a \in \mathsf{A}$ extra structure by choosing $x \in \mathsf{X}$ with $R(x) = a$.
- **"Decorating."** Given $F \colon \mathsf{A} \to \mathbf{Cat}$, we can decorate $a \in A$ with an object $d \in F(a)$.

We want to develop both approaches and then relate them.

There are at least two ways to equip an object of a category A with extra data:

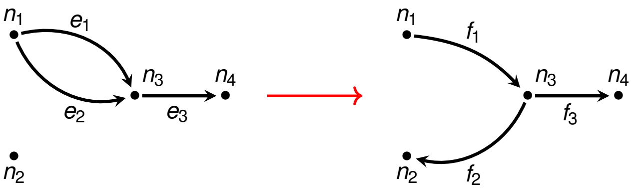
- **"Structuring."** Given a right adjoint $R \colon \mathsf{X} \to \mathsf{A}$, we can give $a \in \mathsf{A}$ extra structure by choosing $x \in \mathsf{X}$ with $R(x) = a$.

- **"Decorating."** Given $F \colon \mathsf{A} \to \mathbf{Cat}$, we can decorate $a \in A$ with an object $d \in F(a)$.

We want to develop both approaches and then relate them.

First let's do "decorating", because it's harder. Consider the example of open graphs.

For any finite set N, there is a category $F(N)$ of graphs on N.



Indeed, there's a lax monoidal pseudofunctor

$$F\colon (\text{FinSet}, +) \to (\mathbf{Cat}, \times)$$

sending each finite set to the category of graphs on that set.

Roughly:

- "pseudofunctor": $F(f) \circ F(g) \cong F(f \circ g)$.
- "lax monoidal": we have $\phi_{N,M}\colon F(N) \times F(M) \to F(N+M)$

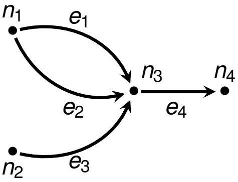
Indeed, there's a lax monoidal pseudofunctor

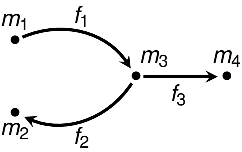
$$F\colon (\mathrm{FinSet}, +) \to (\mathbf{Cat}, \times)$$

sending each finite set to the category of graphs on that set.

Roughly:

- "pseudofunctor": $F(f) \circ F(g) \cong F(f \circ g)$.
- "lax monoidal": we have $\phi_{N,M}\colon F(N) \times F(M) \to F(N + M)$





In general, given a lax monoidal pseudofunctor

$$F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times),$$

a **decorated cospan** is a cospan in A:

$$A \xrightarrow{\;i\;} N \xleftarrow{\;o\;} B$$

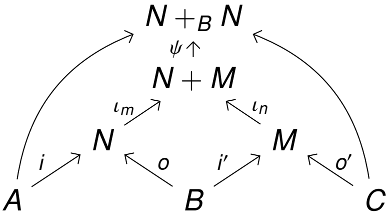
together with a **decoration** $d \in F(N)$.

So, open graphs are decorated cospans where $\mathsf{A} = \mathsf{Set}$ and F maps any set to the category of graphs on that set.

In general, given decorated cospans

$$A \to N \leftarrow B,\ d \in F(N) \qquad N = B \to M \leftarrow C,\ e \in F(M)$$

we compose their underlying cospans by pushout:



and give it the decoration that's the image of (d, e) under this composite:

$$(d, e) \in F(N) \times F(M) \xrightarrow{\phi_{N,M}} F(N + M) \xrightarrow{F(\psi)} F(M +_B N)$$

where $\phi_{N,M}$ comes from F being lax monoidal.

Theorem (B–C–V). Suppose the category A has finite colimits and $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ is a lax monoidal pseudofunctor. Then there is a **decorated cospan double category** $F\mathbb{C}\mathbf{sp}$ where:

- an object is an object of A
- a vertical 1-morphism is a morphism in A
- a horizontal 1-cell is a decorated cospan:

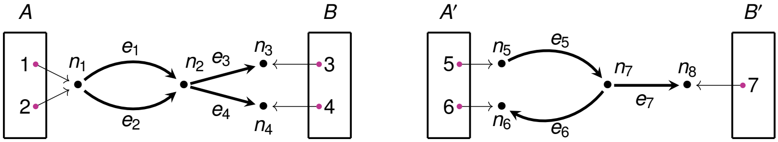
$$A \xrightarrow{i} N \xleftarrow{o} B \qquad d \in F(N)$$

- a 2-morphism is a commuting diagram

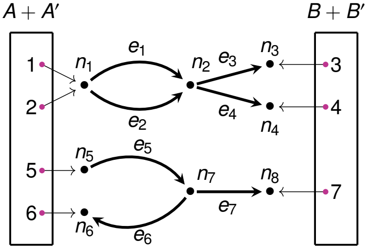
$$\begin{array}{ccccc} A & \xrightarrow{i} & N & \xleftarrow{o} & B \\ \downarrow f & & \downarrow h & & \downarrow g \\ A' & \xrightarrow{i'} & N' & \xleftarrow{o'} & B' \end{array} \qquad \begin{array}{c} d \in F(N) \\ \\ d' \in F(N') \end{array}$$

together with a **decoration morphism** $\tau\colon F(h)(d) \to d'$.

We can also "tensor" open graphs:



by setting them side by side:



This tensor product gives us a *symmetric monoidal* double category of open graphs, thanks to this general result:

Theorem (B–C–V). Suppose the category A has finite colimits and $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ is a symmetric lax monoidal pseudofunctor. Then the decorated cospan double category $F\mathbb{C}\mathbf{sp}$ is symmetric monoidal.

Next let's look at *structured* cospans.

Next let's look at *structured* cospans.

Given a right adjoint

$$R \colon \mathsf{X} \to \mathsf{A}$$

a structured cospan is a diagram in A of this form:

$$A \xrightarrow{\;i\;} R(X) \xleftarrow{\;o\;} B$$

Think of A as a category of objects with "less structure", and X as a category of objects with "more structure".

Next let's look at *structured* cospans.

Given a right adjoint

$$R\colon \mathsf{X} \to \mathsf{A}$$

a structured cospan is a diagram in A of this form:

$$A \xrightarrow{\;i\;} R(X) \xleftarrow{\;o\;} B$$

Think of A as a category of objects with "less structure", and X as a category of objects with "more structure".

For open graphs $\mathsf{A} = \mathsf{Set}, \mathsf{X} = \mathsf{Graph}$ and $R(X)$ is the set of nodes of the graph X.

Equivalently, given a left adjoint

$$L \colon \mathsf{A} \to \mathsf{X}$$

a **structured cospan** is a diagram in X of this form:

$$L(A) \xrightarrow{i} X \xleftarrow{o} L(B)$$

Now we can compose structured cospans by doing pushouts in X.

Equivalently, given a left adjoint

$$L \colon \mathsf{A} \to \mathsf{X}$$

a **structured cospan** is a diagram in X of this form:

$$L(A) \xrightarrow{\;i\;} X \xleftarrow{\;o\;} L(B)$$

Now we can compose structured cospans by doing pushouts in X.

For open graphs $\mathsf{A} = \mathsf{Set}, \mathsf{X} = \mathsf{Graph}$ and $L(A)$ is the graph with A as its set of nodes and no edges.

Theorem (B–C, B–C–V). Suppose A and X have finite colimits and $L\colon \mathsf{A} \to \mathsf{X}$ is a left adjoint. Then there is a symmetric monoidal **structured cospan double category** ${}_L\mathbb{C}\mathbf{sp}(\mathsf{X})$ 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

$$\begin{array}{ccccc} L(A) & \xrightarrow{i} & X & \xleftarrow{o} & L(B) \\ {\scriptstyle L(f)}\downarrow & & \downarrow{\scriptstyle h} & & \downarrow{\scriptstyle L(g)} \\ L(A') & \xrightarrow[i']{} & X' & \xleftarrow[o']{} & L(B') \end{array}$$

Not all decorated cospans can be seen as structured cospans!

Open dynamical systems can be described using decorated cospans, but apparently not using structured cospans. The forgetful functor from dynamical systems to finite sets has no left adjoint **B–C–V).**

This is theoretically interesting — but also of practical importance now that we're using software to manipulate structured and decorated cospans.

When are decorated copans also structured cospans?

When are decorated copans also structured cospans?

For this we must relate the key ingredient for decorated cospans:

$$F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$$

to the key ingredient for structured cospans:

$$R\colon \mathsf{X} \to \mathsf{A}$$

When are decorated copans also structured cospans?

For this we must relate the key ingredient for decorated cospans:

$$F\colon (\mathsf{A},+) \to (\mathbf{Cat},\times)$$

to the key ingredient for structured cospans:

$$R\colon \mathsf{X} \to \mathsf{A}$$

For any pseudofunctor $F\colon \mathsf{A} \to \mathbf{Cat}$ we can use the Grothendieck construction to build a category $X = \int F$ and a functor $R\colon \mathsf{X} \to \mathsf{A}$ which is an "opfibration".

But we're assuming $F\colon (\mathsf{A},+) \to (\mathbf{Cat},\times)$ is a *symmetric lax monoidal* pseudofunctor, and we want R to be a *right adjoint*.

Theorem (Shulman & Moeller–Vasilakopoulou). If A has finite coproducts, these three things correspond to each other:

- symmetric lax monoidal pseudofunctors $F \colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$
- pseudofunctors $F \colon \mathsf{A} \to \mathbf{SymMonCat}$.
- symmetric monoidal opfibrations $R \colon (\mathsf{X}, \otimes) \to (\mathsf{A}, +)$

Example. Take $\mathsf{A} = \mathrm{Set}$ and let $F\colon (\mathrm{Set}, +) \to (\mathbf{Cat}, \times)$ send each set N to the category $F(N)$ of graphs on N. This is lax symmetric monoidal where the laxator

$$\phi_{N,M}\colon F(N) \times F(M) \to F(N + M)$$

sends a pair of graphs to their disjoint union.

Example. Take $\mathsf{A} = \mathsf{Set}$ and let $F\colon (\mathsf{Set}, +) \to (\mathbf{Cat}, \times)$ send each set N to the category $F(N)$ of graphs on N. This is lax symmetric monoidal where the laxator

$$\phi_{N,M}\colon F(N) \times F(M) \to F(N+M)$$

sends a pair of graphs to their disjoint union.

Here $F(N)$ has finite colimits, and becomes symmetric monoidal using $+$. We thus get a pseudofunctor $F\colon \mathsf{A} \to \mathbf{SymMonCat}$. Moreover F factors through $\mathbf{Rex}$, the 2-category of categories with finite colimits.

Example. Take $\mathsf{A} = \mathrm{Set}$ and let $F\colon (\mathrm{Set}, +) \to (\mathbf{Cat}, \times)$ send each set N to the category $F(N)$ of graphs on N. This is lax symmetric monoidal where the laxator

$$\phi_{N,M}\colon F(N) \times F(M) \to F(N+M)$$

sends a pair of graphs to their disjoint union.

Here $F(N)$ has finite colimits, and becomes symmetric monoidal using $+$. We thus get a pseudofunctor $F\colon \mathsf{A} \to \mathbf{SymMonCat}$. Moreover F factors through **Rex**, the 2-category of categories with finite colimits.

Here $\mathsf{X} = \int F$ is the category of graphs, and $R\colon (\mathsf{X}, \otimes) \to (\mathrm{Set}, +)$, sending each graph to its set of nodes, is a symmetric monoidal opfibration.

Theorem (B–C–V). Suppose A has finite colimits and

$$F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$$

is a symmetric lax monoidal pseudofunctor. Suppose the corresponding pseudofunctor

$$F\colon \mathsf{A} \to \mathbf{SymMonCat}$$

factors through **Rex.** Then the symmetric monoidal double categories:

- $F\mathbb{C}\mathbf{sp}$ of decorated cospans

and

- ${}_L\mathbb{C}\mathbf{sp}(\int F)$ of structured cospans

are isomorphic, where $L\colon \mathsf{A} \to \int F$ is a left adjoint of the functor $R\colon \int F \to \mathsf{A}$ given by the Grothendieck construction.

Open problems:

- When are decorated cospan double categories also structured cospan double categories? (Cicala and Vasilakopoulou know a lot about this!)

Open problems:

- When are decorated cospan double categories also structured cospan double categories? (Cicala and Vasilakopoulou know a lot about this!)
- Even when a decorated cospan double category is not a structured cospan double category, sometimes the corresponding *category* is! When does this happen?

Open problems:

- When are decorated cospan double categories also structured cospan double categories? (Cicala and Vasilakopoulou know a lot about this!)
- Even when a decorated cospan double category is not a structured cospan double category, sometimes the corresponding *category* is! When does this happen?
- Can we develop a taxonomy of the *useful* structured and decorated cospan double categories and maps between them? This could be “a universal language of open systems”.

Open problems:

- When are decorated cospan double categories also structured cospan double categories? (Cicala and Vasilakopoulou know a lot about this!)
- Even when a decorated cospan double category is not a structured cospan double category, sometimes the corresponding *category* is! When does this happen?
- Can we develop a taxonomy of the *useful* structured and decorated cospan double categories and maps between them? This could be “a universal language of open systems”.
- Will we create software to do practical things with structured and decorated cospan *double* categories, not just categories?