DOUBLE CATEGORIES OF OPEN SYSTEMS: THE COSPAN APPROACH

JOHN C. BAEZ

ABSTRACT. This is an overview of double categories of 'open systems': systems that can interact with their environment. We focus on the variable sharing paradigm, where we compose open systems by identifying variables. This paradigm is often implemented using structured or decorated cospans. We explain this approach using three main examples: open Petri nets, open dynamical systems, and open Petri nets with rates. We compare the virtues of structured and decorated cospan double categories, and study their common features. We show that any symmetric monoidal structured or decorated cospan double category comes with maps from two simpler double categories: its 'exoskeleton' and its 'outer shell'. Finally, we study the concept of 'hypergraph double category', a kind of double category that should subsume structured and decorated cospans in a common framework for studying open systems in the variable sharing paradigm.

CONTENTS

1. Introduction 1
2. Open systems 3
3. Structured and decorated cospans 7
4. Petri nets 13
5. Dynamical systems 21
6. Petri nets with rates 31
7. The variable sharing paradigm 38
References 52

1. INTRODUCTION

When writing his foundational papers on double categories with Marco Grandis, Robert Paré never expected that their work would be applied to open systems: that is, systems that can interact with their environment. Open systems are fundamental to modern technology, so they have been studied intensively, but still not as much as 'closed' systems, where we neglect the system's interaction with its environment. Some aspects of open systems have only recently been formalized. One is the *composition* of open systems—building larger systems out of smaller parts. Another is *functorial semantics* for open systems, where various kinds of diagrams are used to describe open systems and the maps between them. It turns out that double categories are well suited to both these purposes!

Diagrams of open systems, such as electrical circuit diagrams, can often be seen as loose morphisms in some double category $\mathbb{D}$. Maps between diagrams are then 2-cells in

SCHOOL OF MATHEMATICS, UNIVERSITY OF EDINBURGH, EDINBURGH UK, EH9 3FD
E-mail address: baez@math.ucr.edu.

$\mathbb{D}$. This double category $\mathbb{D}$ serves as a *syntax*. A double functor F from $\mathbb{D}$ to some other double category $\mathbb{E}$ then provides a *semantics* that says what the diagrams 'mean'. Loose morphisms in $\mathbb{E}$ are typically systems of equations of some kind—algebraic equations, differential equations, etc.—with certain variables singled out that can interact with the outside world.

A number of developments were required to enable the double categorical approach to open physical systems. One key step was Grandis and Paré's 1999 paper [48] introducing double categories where composition is strictly associative in one direction (now called the 'tight' direction) and weakly associative in the other (now called the 'loose' direction). By now this concept is the default notion of double category—and the only one we use here.

Another important step was the rise of double categories where a tight morphism $f\colon x \to y$ can be turned into a loose morphism in two ways, its 'companion' $f_*\colon x \nrightarrow y$ and its 'conjoint' $f^*\colon y \nrightarrow x$. While these have their roots in Wood's 'proarrow equipments' [89, 90], the companion and conjoint have universal properties that Paré and others elegantly stated in double categorical language [38, 49]. Around 2007, Shulman [82] studied double categories with companions and conjoints under the name 'framed bicategories'. In 2010 he gave them another name: 'fibrant double categories' [83], because they are the same as double categories $\mathbb{D}$ where the source and target maps going from the category $\mathbb{D}_1$ of loose morphisms and 2-cells to the category $\mathbb{D}_0$ of objects and tight morphisms combine to give a functor $\mathbb{D}_1 \to \mathbb{D}_0 \times \mathbb{D}_0$ that is a Grothendieck fibration.

A crucial step toward applying these ideas to open systems was the introduction of topological quantum field theories described as functors. This expanded Lawvere's idea of functorial semantics [61] in a radically non-cartesian direction, and it emphasized the important of compact closed categories, and higher analogues of these, for understanding physical systems. For reviews of this line of work see [15, 21, 60].

When these ideas came together, it was only a matter of time before double categories were used to study open systems. But this general idea can be made more precise in numerous ways. For example, there are various choices of what it means to compose open systems. In the 'input-output paradigm', when we compose two open systems, variables of one can affect variables of the other while not being directly affected by them. In the 'variable sharing paradigm', we instead *identify* variables of one system with variables of another.

In recent years Libkind has formalized both these paradigms and begun to unify them [63], Myers has studied the input-output paradigm using double categories [70], and together they have formalized the very concept of 'paradigm' [64, 71]. To limit the scope of this paper, we only discuss the variable sharing paradigm, and emphasize the approach to this paradigm based on cospans. We illustrate this using three examples of open systems: Petri nets, dynamical systems described by systems of first-order differential equations, and finally Petri nets with rates. Thus, we omit or only briefly touch upon many other examples of the variable sharing paradigm, including:

- electrical circuits [7, 10, 12],
- Markov processes [8, 13],
- bond graphs [30, 31],
- signal flow diagrams [11, 23, 31, 44],
- stock-flow diagrams and system structure diagrams [16, 17],
- gene regulatory networks and causal loop diagrams [1, 6, 17],
- thermodynamics [18, 65],
- classical mechanics [22, 65].

In Section 2 we explain open systems and various ways to formalize them, leading up the double categorical approach. In Section 3 we introduce two ways to construct double categories of open systems: structured and decorated cospans. We discuss the relative merits of these approaches, and raise some questions about the relation between them. In Section 4 we illustrate structured cospan double categories and maps between them using the example of open Petri nets. In Section 5 we illustrate decorated cospan categories using the example of open dynamical systems. We look at some applications to classical mechanics, and revisit the question of when to use structured and when to use decorated cospans. In Section 6, we illustrate maps between decorated cospans double categories using the map sending open Petri nets with rates to open dynamical systems. We also look at applications of structured and decorated cospans to software for modeling inthe field of public health.

In Section 7, we conclude with a deeper investigation of the variable sharing paradigm. We find that some structures important in topological quantum field theory, such as commutative Frobenius monoids, also appear in the variable sharing paradigm. Indeed, in any symmetric monoidal double category of structured or decorated cospans, every object is a *categorified version* of a commutative Frobenius monoid. This fact expresses the laws obeyed by the operations of bending, splitting and joining wires—where 'wires' should be taken quite generally as routes along which variables can affect one another. In topological quantum field theory, the same laws appear in the description of spacetime itself [15, 59].

1.1. **Conventions.** In this paper, we use a sans-serif font like C for categories, boldface like $\mathbf{B}$ for bicategories or 2-categories, and blackboard bold like $\mathbb{D}$ for double categories. For double categories with names having more than one letter, such as $\mathbb{C}\mathbf{sp}(\mathsf{X})$, only the first letter is in blackboard bold. In this paper, 'double category' means 'pseudo double category'. In a double category 'tight' morphisms are drawn vertically and their composition is strictly associative, while 'loose' morphisms are drawn horizontally and their composition is associative up to coherent isomorphism. We use $(\mathsf{C}, \otimes)$ to stand for a monoidal or perhaps symmetric monoidal category with $\otimes$ as its tensor product.

2. Open systems

An 'open system' is a system that we treat as interacting with the outside world—as opposed to a 'closed system', where interactions with the outside world are absent or at least neglected. To a mathematician this may seem hopelessly vague, but physicists and engineers have various fairly precise frameworks for discussing 'systems' of various kinds, and in many of these frameworks one can set up a distinction between open and closed systems. For example, in the realm of electrical circuits, a circuit with 'terminals' or 'ports' exposed to the outside world is an open system, while one without these is a closed system.

A striking fact is that so far, more mathematical labor has gone into proving results about *closed* systems than open ones. The main reason is that closed systems are easier to understand. Thus, there is a lot of interesting new territory yet to be explored in the realm of open systems. I believe category theory can greatly assist this exploration, and that it will also benefit from new ideas discovered in this exploration. The reason is that category theory provides many tools for discussing open systems.

We can combine two closed systems by 'setting them side by side', letting each do its own thing, neither influencing nor being influenced by the other. In this situation we can easily reduce the study of the composite system to the study of each separate part. Open systems can be composed in more interesting ways, which create interactions between the

two parts. For example, we can take two electrical circuits and attach the terminals of one to the terminals of the other, forming a larger circuit. More generally, we can think of any open system as having one or more 'interfaces', through which it can interact with the outside world, and to compose open systems we attach them along an interface. This is an idealization, but a useful one. Given this, several ways of formalizing a chosen class of open systems spring to mind.

One approach is to use a symmetric monoidal category where:

- an object is an 'interface',
- a morphism $F\colon x \to y$ is an 'open system' with 'left interface' x and 'right interface' y,
- composing morphisms $F\colon x \to y$ and $G\colon y \to z$ is attaching the left interface of F to the right interface of G, obtaining an open system $G \circ F\colon x \to z$,
- tensoring two morphisms $F\colon x \to y$ and $F'\colon x' \to y'$ is 'setting two open systems side by side', giving an open system $F \otimes F'\colon x \otimes x' \to y \otimes y'$.

There are many questions that can be raised about this approach. First, why should an open system have just two interfaces? There is no reason it must. Luckily, this can be addressed by the fact that we are working in a symmetric monoidal category: two interfaces x and x' sitting side by side can be reinterpreted as a single interface $x \otimes x'$. Thus, an open system f with any number of left interfaces $x_1, \ldots, x_m$ and any number of right interfaces $y_1, \ldots, y_n$ can be interpreted as a morphism

$$f\colon x_1 \otimes \cdots \otimes x_m \to y_1 \otimes \cdots \otimes y_n.$$

We can also think of this as having a single left interface $x = x_1 \otimes \cdots \otimes x_m$ and a single right interface $y = y_1 \otimes \cdots \otimes y_n$.

Second, why should there be a distinction between left and right interfaces, and why can we only compose systems by attaching the left interface of one to the right interface of the other? Again, we do not need to take this approach. It is natural for systems with a well-defined 'input' and 'output'—but these are more common among engineered systems than naturally occuring ones. The language of 'input' and 'output' suggests that when we compose open systems $f\colon x \to y$ and $g\colon y \to z$ to create an open system $g \circ f\colon x \to z$, the behavior of f affects g but not conversely. This is a very useful simplifying assumption. But in the natural world, this assumption is at best an approximation that holds under very limited circumstances. A widely applicable principle in physics, called the 'principle of reciprocity', says that if one system can affect another, then the second system can also affect the first.

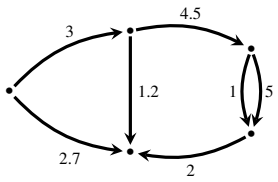
It is quite interesting to see how engineered systems evade the principle of reciprocity without actually violating it. This makes for a good puzzle. Flipping a switch can turn on a light; why is the behavior of the light apparently unable to affect the position of the switch? The click of the switch as it snaps into position is a clue. The switch not only turns the light on or off: it also releases energy in the form of sound waves as it settles into a low-energy state, either 'on' or 'off'. The principle of reciprocity says that we could reverse the whole scenario and have the behavior of the light *and the sound waves in the room* control the switch. However, it is very difficult to focus sound waves on a switch in the necessary way, so this never happens in practice. A similar story is at work in other physical systems that appear to have inputs and output.

In any event, we want a formalism that lets us study open systems that have a rigid distinction between input and output, but also those where the distinction between left and

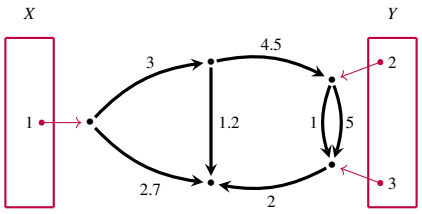
right interface is purely conventional. We can study *both* using symmetric monoidal categories. When desired, we can eliminate the distinction between input and output in several ways. One is to work with a symmetric monoidal dagger-category: here any morphism $f\colon x \to y$ gives a morphism $f^\dagger\colon y \to x$. Another is to work with a compact closed category: here every object x has a dual x^*, and any morphism $f\colon x \to y$ gives a morphism $f^*\colon y^* \to x^*$, as well as a morphism $I \to x^* \otimes y$ and a morphism $x \otimes y^* \to I$. Another way is to do both, and work with a 'dagger compact category'. All these options, and others, have been extensively studied [21, 54, 80, 81].

There are further features of the variable-sharing approach to open systems that call for a more flexible approach based on operads, particularly the operad of undirected wiring diagrams [84, 91]. Operads easily allow for more composition patterns more general than the 'end-to-end' composition of morphisms in categories. But an even more urgent issue appears as soon as we consider an example: say, electrical circuits.

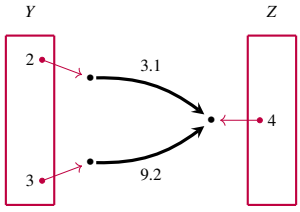
For simplicity, consider electrical circuits made using only resistors. We can describe such a circuit as a finite graph where each edge is labeled by a positive real number called its 'resistance':



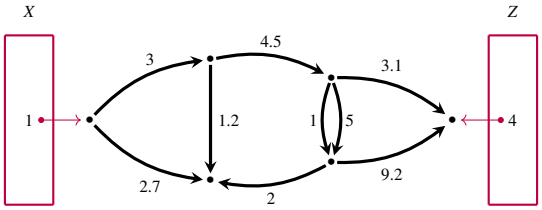
(We apologize to electrical engineers for not decorating the edges with symbols indicating resistors.) We can treat such a circuit as an *open* system by choosing some nodes to be inputs and some to be outputs. Even better, for mathematicians at least, is to choose two finite sets X and Y equipped with maps to the circuit's set of nodes:



We can think of this as a morphism from X to Y. We can compose this with a morphism from Y to Z:



and the result is this:



Nothing profound is going on here: we are simply gluing together two open electrical circuits to get a new one. But there is something worth noting. This process of gluing is really a pushout, and pushouts are only defined up to canonical isomorphism. Thus, we cannot expect composition of open circuits to be strictly associative, only associative up to isomorphism. Thus, to create a category with this composition, we either need to take isomorphism classes of open circuits as morphisms, or choose the pushouts very carefully—a drastic measure that scarcely bears thinking about.

Working with isomorphism classes is not good here, because one cannot point to a specific node or edge in an *isomorphism class* of graphs: for that, one needs an actual graph. There is a general lesson here: instead of a category with open systems as morphisms, we should seek a bicategory—or better yet, a double category!

In fact Courser [27] had been trying to construct symmetric monoidal bicategories of open systems. But the coherence laws for symmetric monoidal bicategories [50, 67, 85] are complicated and tiring to check directly. He realized that the easiest approach was to use a result of Shulman [83]: any fibrant symmetric monoidal double category gives a symmetric monoidal bicategory. Later it became clear that double categories are closer to what we actually want, since in addition to having open systems as loose morphisms between interfaces, which compose in a weakly associative manner, they have tight morphisms as *maps* between interfaces, which compose in a strictly associative manner.

Thus, a better approach to open systems is to use a symmetric monoidal double category where:

- An object is an 'interface'.
- A loose morphism $F\colon x \nrightarrow y$ is an 'open system' with 'input interface' x and 'output interface' y.
- Composing loose morphisms $F\colon x \nrightarrow y$ and $G\colon y \nrightarrow z$ is attaching the output interface of F to the input interface of G, obtaining an open system $G \circ F\colon x \to z$.
- A tight morphism $f\colon x \to y$ is a 'map between interfaces'.
- Composing tight morphisms is composing maps between interfaces.
- A 2-cell

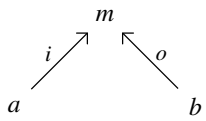
$$\begin{array}{ccc} x & \xrightarrow{F} & y \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ x' & \xrightarrow{G} & y' \end{array}$$

  is a 'map between open systems' from F to G.
- Tensoring two loose morphisms $F\colon x \nrightarrow y$ and $F'\colon x' \nrightarrow y'$ is 'setting two open systems side by side', giving an open system $F \otimes F'\colon x \otimes x' \nrightarrow y \otimes y'$.
- Tensoring two tight morphisms is 'setting two maps of interfaces side by side'.
- Tensoring two 2-cells is 'setting two maps of open systems side by side'.

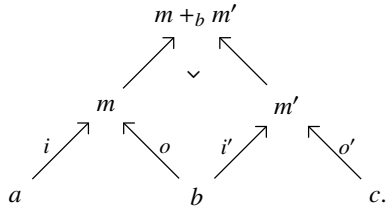
Next we describe two methods of defining such double categories: structured and decorated cospans. It will often be convenient to state theorems in two parts: first one for double categories, then one for symmetric monoidal double categories. In practice, the symmetric monoidal version is always the more useful one—but it relies on the simpler version that ignores the symmetric monoidal structure.

3. Structured and decorated cospans

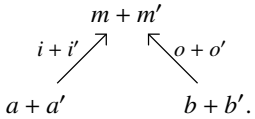
Experience has shown that many open systems are nicely modeled using cospans [28, 43, 77]. A cospan in a category A is a diagram of this form:



We call m the **apex**, a and b the **feet**, and i and o the **legs** of the cospan. The apex is the system itself. The feet are 'interfaces' through which the system can interact with the outside world. The legs say how the interfaces are included in the system. If the category A has pushouts, we can compose cospans in A: this describes the operation of attaching two open systems together in series by identifying one interface of the first with one of the second:

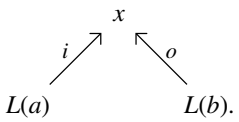


If A also has coproducts, we can also 'tensor' cospans: this describes setting open systems side by side, in parallel:



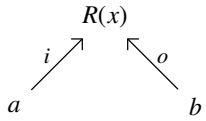
However, we often want the system itself to have more structure than its interfaces. We know of two main ways to do this: structured and decorated cospans.

3.1. **Structured cospans.** Given a functor $L\colon \mathsf{A} \to \mathsf{X}$, an L-**structured cospan** is a cospan in X whose feet come from a pair of objects in A:



Very often A is some category of objects with 'less structure', like finite sets, while X is some category of objects with 'more structure', like finite graphs, and $L\colon \mathsf{A} \to \mathsf{X}$ is left adjoint to some functor $R\colon \mathsf{X} \to \mathsf{A}$ that forgets the extra structure. Thus we often think of an L-structured cospan as consisting of a 'system' $x \in \mathsf{X}$ and two 'interfaces' $a, b \in \mathsf{A}$ that have less structure than the system, mapped into the system via i and o.

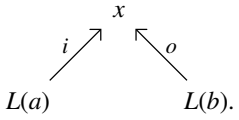
When L is left adjoint to some functor R, as often the case, an L-structured cospan is equivalent to a diagram



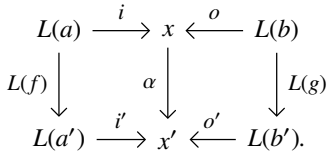
together with a specific choice of $x \in \mathsf{X}$, and this is often a more intuitive way to think about it. However, the description using L works better technically, since it lets us compose structured cospans as long as X has pushouts.

Theorem 3.1. *Let $L\colon \mathsf{A} \to \mathsf{X}$ be a functor and let X be a category with pushouts. Then there is a fibrant double category ${}_L\mathbb{C}\mathbf{sp}$, the* **double category of L-structured cospans***, in which*

- *an object is an object of A,*
- *a tight morphism is a morphism of A,*
- *a loose morphism from a to b is an L-structured cospan*



- *a 2-cell is a* **map of L-structured cospans***, that is, a commutative diagram in X of the form*



Composition of tight morphisms is composition in A. Composition of loose morphisms and 2-cells is done using pushouts in X.

Proof. See [7, Thm. 2.3] or [28, Thm. 3.2.1], and [74, Prop. 2.2] for the fibrancy. □

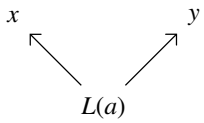
In practice we almost always seem to work with symmetric monoidal double categories of structured cospans—and in fact, cocartesian ones. Dualizing Aleiferi's work on cartesian double categories [2], we define a **cocartesian** double category $\mathbb{D}$ to be a cocartesian object in the 2-category of double categories. This implies that the category $\mathbb{D}_0$ of objects

and tight morphisms and the category $\mathbb{D}_1$ of loose morphisms and 2-cells are categories with finite coproducts, and the source, target, composition, and identity-assigning maps preserve these.

Theorem 3.2. *Let $L\colon \mathsf{A} \to \mathsf{X}$ be a functor preserving finite coproducts, where A has finite coproducts and X has finite colimits. Then ${}_L\mathbb{C}\mathbf{sp}$ naturally has the structure of a cocartesian double category. It also becomes a symmetric monoidal double category, where the symmetric monoidal structure is defined using finite coproducts in A and X.*

Proof. This is a combination of [28, Thm. 3.2.3] , [7, Thm. 3.9], [9, Thm. 3.2.1], and [74, Thm. 2.3]. □

It is worth saying a bit about the hypotheses in Theorem 3.2. They have been pared down to almost the minimum necessary. In practice $L\colon \mathsf{A} \to \mathsf{X}$ is often a left adjoint functor between categories with finite colimits, which is more than enough for the theorem to apply. However, in one application [6, Thm. 7.6] it seemed necessary to pare down the hypotheses even more. To get the conclusions of Theorem 3.2, we do not need X to have all finite colimits. It is enough that it have finite coproducts and have pushouts for diagrams of this form:



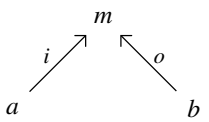
since these are the only pushouts required to compose structured cospans.

Finally, a word about history. Structured cospans were first discovered in 2007 by Fiadeiro and Schmitt [41]. In 2015 Paré gave a talk about the dual concept, structured spans, which he called 'superspans' [73]. Completely ignorant of these earlier developments, Courser and the author [7] reinvented structured cospans around 2020, even giving them the same name that Fiaideiro and Schmitt used. This is an example of how a mathematical concept can be invented repeatedly, but only catch on when its applications become clear.

3.2. **Decorated cospans.** Structured cospans handle a large class of cospans where the apex has more structure than the feet—but apparently not all. As mentioned, structured cospans assume we fix a category A of 'interfaces' and a category X of 'systems', together with a functor $L\colon \mathsf{A} \to \mathsf{X}$ that gives a standard way to turn an interface into a system. To get a double category of structured cospans, we need X to have pushouts. But we shall see examples where the category of systems does not have pushouts.

We can get around this problem using decorated cospans [9, 42]. Here we start with a category A with finite colimits and a pseudofunctor $F\colon \mathsf{A} \to \mathbf{Cat}$. We again think of objects of A as 'interfaces'. But now, for each interface $m \in \mathsf{A}$, we have a category $F(m)$ of ways to equip it with extra data making it into a system. We call a choice of this extra data $d \in F(m)$ a **decoration** of m.

In this approach we define an open system to be an F-**decorated cospan**: a cospan in A together with a decoration of its apex. We write an F-decorated cospan in this way:

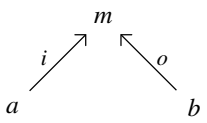


$d \in F(m)$.

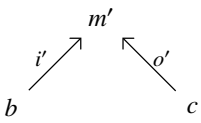
To compose decorated cospans, we need to equip F with the structure of a lax monoidal pseudofunctor from $(\mathsf{A}, +)$ to $(\mathbf{Cat}, \times)$. The key ingredient here beyond F itself is the 'laxator', which gives for each pair of objects $m, m' \in \mathsf{A}$ a functor

$$\phi_{m,m'} \colon F(m) \times F(m') \to F(m + m').$$

Given a composable pair of F-decorated cospans

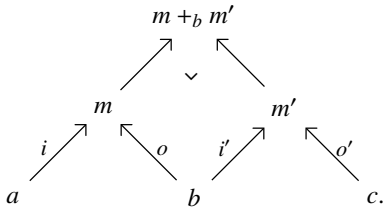


$d \in F(m)$



$d' \in F(m')$

we define their composite to be

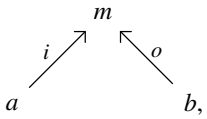


$F(j)\,(\phi_{m,m'}(d, d')) \in F(m +_b m')$

Here the cospans are composed using a pushout in A, while the decoration is defined using the laxator and F applied to $j \colon m + m' \to m +_b m'$, the canonical map from the coproduct to the pushout. The choice of laxator gives a fair amount of flexibility in how the new cospan is decorated—but the laxator must obey some laws that guarantee that this prescription gives a double category [9, App. A.1].

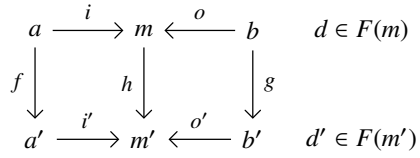
Theorem 3.3. *Let* A *be a category with finite colimits and let* $F \colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ *be a lax monoidal pseudofunctor. Then there is a double category* $F\mathbb{C}\mathbf{sp}$*, the* **double category of** F**-decorated cospans***, in which*

- *an object is an object of* A*,*
- *a tight morphism is a morphism of* A*,*
- *a loose morphism is an* F*-decorated cospan*



$d \in F(m)$

- *a 2-cell is a* **map of** F**-decorated cospans***, that is, a commutative diagram in* A *of the form*



  together with a **decoration morphism** $\tau \colon F(h)(d) \to d'$ *in* $F(m')$*,*
- *composition of tight morphisms and vertical composition of 2-cell is done using composition in* A*,*
- *composition of loose morphisms is done as described above,*

- *horizontal composition of 2-cells is done as described in* [9, Thm. 2.1].

Proof. See [9, Thm. 2.1]. □

Equipping F with some more structure, the double category of F-decorated cospans becomes symmetric monoidal:

Theorem 3.4. *Let* A *be a category with finite colimits and let* $F\colon (\mathsf{A},+) \to (\mathbf{Cat},\times)$ *be a symmetric lax monoidal pseudofunctor with laxator* $\phi_{a,b}\colon F(a)\times F(b) \to F(a+b)$*. Then the double category* $F\mathbb{C}\mathbf{sp}$ *becomes symmetric monoidal, where the tensor product*

- *of two objects* a *and* b *is their coproduct* $a+b$ *in* A*,*
- *of two tight morphisms* $f\colon a_1 \to a_2$ *and* $f'\colon a_1' \to a_2'$ *is* $f+f'\colon a_1+a_1' \to a_2+a_2'$ *in* A*,*
- *of two loose morphisms* $(a \to m \leftarrow b, d \in F(m))$ *and* $(a' \to m' \leftarrow b', d' \in F(n))$ *is:*

$$\begin{array}{ccccc} & & m+m' & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ a+a' & & & & b+b' \end{array} \qquad \phi_{m,m'}(d,d') \in F(m+m')$$

- *of two 2-cells* α *and* β *is:*

$$\begin{array}{ccccc} a_1 & \xrightarrow{i_1} & m_1 & \xleftarrow{o_1} & b_1 \\ {\scriptstyle f}\downarrow & & {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle g} \\ a_2 & \xrightarrow[i_2]{} & m_2 & \xleftarrow[o_2]{} & b_2 \end{array} \quad \otimes \quad \begin{array}{ccccc} a_1' & \xrightarrow{i_2'} & m_1' & \xleftarrow[o_2]{} & b_1' \\ {\scriptstyle f'}\downarrow & & {\scriptstyle h'}\downarrow & & \downarrow{\scriptstyle g'} \\ a_2' & \xrightarrow[i_2']{} & m_2' & \xleftarrow[o_2']{} & b_2' \end{array} \quad =$$

$$\tau_\alpha\colon F(h)(d_1) \to d_2 \text{ in } F(m_2) \qquad \tau_\beta\colon F(h')(d_1') \to d_2' \text{ in } F(m_2')$$

$$\begin{array}{ccccc} a_1+a_1' & \xrightarrow{i_1+i_2'} & m_1+m_1' & \xleftarrow{o_1+o_1'} & b_1+b_1' \\ {\scriptstyle f+f'}\downarrow & & {\scriptstyle h+h'}\downarrow & & \downarrow{\scriptstyle g+g'} \\ a_2+a_2' & \xrightarrow[i_2+i_2']{} & m_2+m_2' & \xleftarrow[o_2+o_2']{} & b_2+b_2' \end{array}$$

$$\tau_{\alpha\otimes\beta}\colon F(h+h')(\phi_{m_1,m_1'}(d_1,d_1')) \to \phi_{m_2,m_2'}(d_2,d_2') \text{ in } F(d_2,d_2')$$

with decoration morphism $\tau_{\alpha\otimes\beta}$ *constructed functorially from* τ_α *and* τ_β*.*

Proof. See [9, Thm. 2.2], which also includes the details of how we construct the decoration morphism $\tau_{\alpha\otimes\beta}$. □

3.3. **Structured versus decorated cospans.** When should we use structured cospans, and when should we use decorated cospans? Structured cospans are usually easier to work with, because they are defined using 1-categorical data: a functor $L\colon \mathsf{A} \to \mathsf{X}$ with some properties. Decorated cospans are generally defined using more complicated 2-categorical data: a lax monoidal pseudofunctor $F\colon \mathsf{A} \to \mathbf{Cat}$. Furthermore, decorated cospans can often be reformulated as structured cospans. These observations favor structured cospans—and thus, these are more widely used in software [26, 52].

There is an important exception. When the categories $F(a)$ for $a \in \mathsf{A}$ are all discrete, we can think of them as sets and treat F as a lax monoidal functor $F\colon \mathsf{A} \to \mathsf{Set}$. Then 2-categorical concepts are not needed, and the theory of decorated cospans simplifies dramatically. This in fact was Fong's original approach to decorated cospans [42, 43]. Further,

as we shall see, some examples of this sort seem to give decorated cospans that cannot be reformulated as structured cospans.

When are all the categories $F(a)$ discrete? This happens precisely when the opfibration $U\colon \int F \to \mathsf{A}$ is discrete, where $\int F$ is defined using the Grothendieck construction. An object of $\int F$ is a pair (a, d) consisting of an object $a \in \mathsf{A}$ and a decoration $d \in F(a)$. The opfibration $U\colon \int F \to \mathsf{A}$ forgets the decoration. When U is a discrete opfibration, it means that for any $d \in F(a)$ there is a unique way to equip any morphism $h\colon a \to a'$ in A with a decoration morphism $F(h)(d) \to d'$. This is an unusual situation. But we shall see an important example in Section 5: dynamical systems.

Since structured cospans are usually easier to work with than decorated cospans, another question naturally arises: when are we *forced* to use decorated cospans? The dynamical systems just mentioned appear to be an example. However, there is not yet an airtight proof.

We do know a general result on when decorated cospan double categories *are* equivalent to structured cospan double categories [9]. Begin with the data needed to construct a symmetric monoidal decorated cospan double category, as in Theorem 3.4. That is, suppose A has finite colimits and $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ is a symmetric lax monoidal pseudofunctor. It turns out that F can naturally be promoted to a pseudofunctor $F\colon \mathsf{A} \to \mathbf{SymMonCat}$, where **SymMonCat** is the 2-category with

- symmetric monoidal categories as objects,
- strong symmetric monoidal functors as morphisms,
- monoidal natural transformation as 2-morphisms.

Let **Rex** be the 2-category of with

- categories with chosen finite colimits as objects,
- functors preserving finite colimits as morphisms,
- natural transformations as 2-morphisms.

There is an evident pseudofunctor $\mathbf{Rex} \to \mathbf{SymMonCat}$. Now suppose that $F\colon \mathsf{A} \to \mathbf{SymMonCat}$ factors, as a pseudofunctor, through **Rex**: this is the key hypothesis. Then one can show that the opfibration $U\colon \int F \to \mathsf{A}$ is a right adjoint. From its left adjoint $L\colon \mathsf{A} \to \int F$ we can construct a structured cospan double category ${}_L\mathbb{C}\mathbf{sp}(\mathsf{X})$ by taking

$$\mathsf{X} = \int F.$$

And in fact, under the hypotheses given, the structured cospan double category ${}_L\mathbb{C}\mathbf{sp}(\mathsf{X})$ is not only equivalent but isomorphic to the decorated cospan double category $F\mathbb{C}\mathbf{sp}$.

In short:

Theorem 3.5. *Let* A *be a category with finite colimits and* $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ *a symmetric lax monoidal pseudofunctor. Suppose the resulting pseudofunctor* $F\colon \mathsf{A} \to \mathbf{SymMonCat}$ *factors through* **Rex**. *Then* $U\colon \int F \to \mathsf{A}$ *has a left adjoint L. Furthermore, the structured cospan double category* ${}_L\mathbb{C}\mathbf{sp}$ *is isomorphic as a symmetric monoidal double category, to the decorated cospan double category* $F\mathbb{C}\mathbf{sp}$.

Proof. This is [9, Thm. 4.1]. □

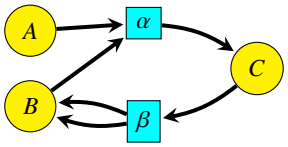
This lets us see certain structured cospans as decorated cospans. The converse question is also interesting: when is a structured cospan double category equivalent to a decorated cospan double category? We believe this is true under certain conditions that let us pass from the functor $L\colon \mathsf{A} \to \mathsf{X}$ to an appropriate pseudofunctor $F\colon \mathsf{A} \to \mathbf{Cat}$. For details, see [9, Sec. 7]. However, the program outlined there has not yet been completed. Thus, we pose this challenge:

Challenge 3.6. *Find necessary and/or sufficient conditions for a symmetric monoidal double category to be equivalent to some decorated cospan double category, and vice versa.*

Next we turn from these rather abstruse issues to something simpler: an example of how we *use* structured cospans.

4. Petri nets

Here is a Petri net:



We can use Petri nets to describe processes where collections of things move between the yellow circles, called 'places', by going through the aqua boxes, called 'transitions'. Petri nets are widely used as models of systems in engineering and computer science [47, 76]. One of the simplest examples of the resource sharing paradigm is the theory of open Petri nets [19], which lets us build Petri nets out of smaller pieces. We can construct a double category with open Petri nets as loose morphisms using the theory of structured cospans. There are many possible semantics for open Petri nets, but we shall only describe one, where 'tokens' move from place to place via transitions. This gives an excuse for studying maps between structured cospan double categories. We also take a look at how to use the 2-cells in double categories of open systems.

While various subtly different definitions are useful [14], here we take a **Petri net** to be a pair of finite sets S and T and functions $s, t \colon T \to \mathbb{N}[S]$. Here S is the set of **places**, T is the set of **transitions**, and $\mathbb{N}[S]$ is the underlying set of the free commutative monoid on S. Thus, elements of $\mathbb{N}[S]$ are finite formal sums of species, and each transition goes from one such formal sum to another. For example, the source of the transition α above is $\mathrm{A} + \mathrm{B}$, because this transition has one arrow coming into it from A and one from B. The target of α is is C, since it has one arrow going out to C. We can summarize all this information by writing

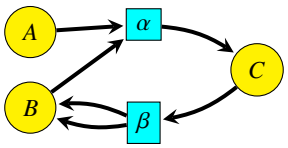
$$\mathrm{A} + \mathrm{B} \xrightarrow{\alpha} \mathrm{C}.$$

Similarly, we can write the other transition as

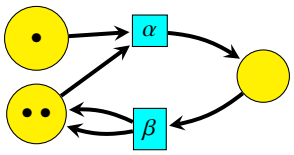
$$\mathrm{C} \xrightarrow{\beta} 2\mathrm{B}.$$

Beware: α and β are not morphisms in a category! They are just transitions in a Petri net. But next we shall see how a Petri net can generate a symmetric monoidal category, with the transitions giving morphisms.

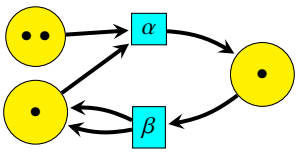
To do this, we define a **marking** of a Petri net $s, t \colon T \to \mathbb{N}[S]$ to be an element of $\mathbb{N}[S]$. Thus, it assigns to each place a natural number. We think of this as specifying a number of **tokens** in each place. We commonly draw these tokens as dots in the yellow circles that represent places. For example, given this Petri net:



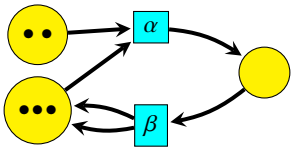
the marking $A + 2B$ would be written as follows:



We change the marking using transitions. For example one token of type A and one of type B can enter the transition α, and a token of type C can come out, giving this marking:



Then the token of type C can enter the transition β, and two of type B can come out, giving this marking:



It is natural to think of these ways of changing a marking as morphisms in a category where markings are objects. This is a symmetric monoidal category, but of a very strict sort: the tensor product commutes on the nose! The reason is that markings form a commutative monoid, namely the free commutative monoid on the set of places.

A **commutative monoidal category** is a symmetric monoidal category $(\mathsf{C}, \otimes)$ such that the associators $\alpha_{a,b,c} \colon (a \otimes b) \otimes c$, unitors $\lambda_a \colon I \otimes a \to a$, $\rho \colon a \otimes I \to a$, and even the symmetry isomorphisms $\sigma_{a,b} \colon a \otimes b \to b \otimes a$ are all identity morphisms. Thus, for all objects a and b and morphisms f and g in C we have

$$a \otimes b = b \otimes a \text{ and } f \otimes g = g \otimes f.$$

We let CMC be the category whose objects are commutative monoidal categories and whose morphisms are strict symmetric monoidal functors.

Another useful way to think of CMC is as the category of categories internal to the category of commutative monoids. In this viewpoint, a commutative monoidal category has a commutative monoid of objects and a commutative monoid of morphisms, and all the category operations are monoid homomorphisms.

We can formalize the process of turning a Petri net $P = (s, t \colon T \to \mathbb{N}[S])$ into a commutative monoidal category FP as follows. We take the commutative monoid of objects $\mathrm{Ob}(FP)$ to be the free commutative monoid $\mathbb{N}[S]$. Note that element of $\mathbb{N}[S]$ are markings of P. We construct the commutative monoid of morphisms $\mathrm{Mor}(FP)$ as follows. First we generate morphisms recursively, starting from the transitions of P:

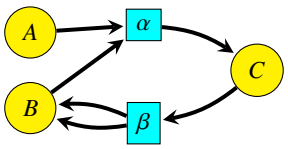
- for every transition $\tau \in T$ we include a morphism $\tau \colon s(\tau) \to t(\tau)$;
- for any object a we include a morphism $1_a \colon a \to a$;
- for any morphisms $f \colon a \to b$ and $g \colon a' \to b'$ we include a morphism denoted $f + g \colon a + a' \to b + b'$ to serve as their tensor product;
- for any morphisms $f \colon a \to b$ and $g \colon b \to c$ we include a morphism $g \circ f \colon a \to c$ to serve as their composite.

Then we mod out by an equivalence relation that imposes the laws of a commutative monoidal category, obtaining the commutative monoid $\mathrm{Mor}(FP)$. The rest of the category structure on FP is straightforward.

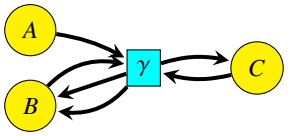
We can extend F to a functor from Petri nets to commutative monoidal categories. Indeed, there is a category Petri where Petri nets are objects and a morphism from the Petri net $s, t\colon T \to \mathbb{N}[S]$ to the Petri net $s', t'\colon T' \to \mathbb{N}[S']$ is a pair of functions $f\colon S \to S', g\colon T \to T'$ such that the following diagrams commute:

$$\begin{array}{ccc} T & \xrightarrow{s} & \mathbb{N}[S] \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \mathbb{N}[g]} \\ T' & \xrightarrow[s']{} & \mathbb{N}[S'] \end{array} \qquad \begin{array}{ccc} T & \xrightarrow{t} & \mathbb{N}[S] \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \mathbb{N}[g]} \\ T' & \xrightarrow[t']{} & \mathbb{N}[S'], \end{array}$$

where $\mathbb{N}[g]$ acts to map any finite sum $\sum_i \sigma_i$ with $\sigma_i \in S$ to $\sum_i g(\sigma_i)$. For example, there is a morphism from this Petri net:



to this one:



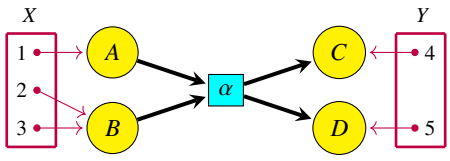
mapping α and β to γ and acting as the identity on the places A, B and C. Then, there is a functor

$$F\colon \mathsf{Petri} \to \mathsf{CMC}$$

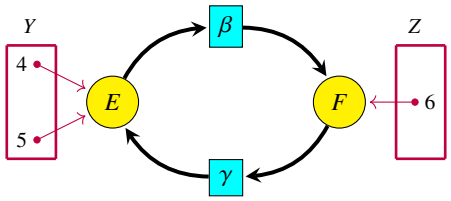
sending any Petri net P to the free commutative monoidal category FP that we have already described. Moreover, this is a left adjoint [66, Thm. 5.1].

We should think of F as a 'semantics' for Petri nets, saying what they can 'mean'. In this semantics the 'meaning' of a Petri net P is the free commutative monoidal category FP describing how tokens can move around from place to place. Thus we call F the **token semantics**.

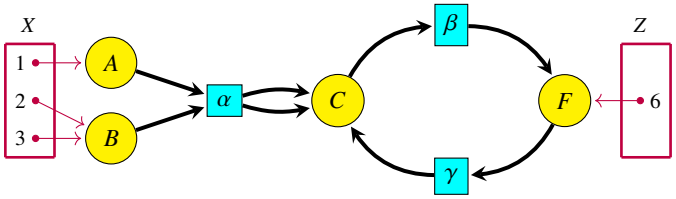
4.1. **Open Petri nets.** We now turn to 'open' Petri nets, and construct a double category whose loose morphisms are open Petri nets. This lets us build Petri nets out of smaller open parts, which allows us to study Petri nets and their semantics compositionally. For example, here is a picture of an open Petri net from a finite set X to a finite set Y:



We shall think of it as a loose morphism in a double category and write it as $P\colon X \nrightarrow Y$ for short. Given another open Petri net $Q\colon Y \nrightarrow Z$:



we can compose them and get an open Petri net $Q \circ P\colon X \nrightarrow Z$:



To formalize this, first note that there is a functor $R\colon \mathsf{Petri} \to \mathsf{FinSet}$ sending any Petri net to its set of places. This has a left adjoint $L\colon \mathsf{FinSet} \to \mathsf{Petri}$ sending any finite set S to the Petri net with S as its set of places and no transitions [19, Lem. 11]. Since both FinSet and Petri have finite colimits and L preserves them, Theorem 3.2 yields a symmetric monoidal double category ${}_L\mathbb{C}\mathbf{sp}$ in which:

- an object is a finite set,
- a tight morphism is a function,
- a loose morphism is an **open Petri net**, meaning a cospan in Petri of this form:

$$\begin{array}{ccccc} & & P & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ L(X) & & & & L(Y) \end{array}$$

- a 2-cell is a **map of open Petri nets**, meaning a commutative diagram in Petri of this form:

$$\begin{array}{ccccc} L(X) & \xrightarrow{i} & P & \xleftarrow{o} & L(Y) \\ {\scriptstyle L(f)}\downarrow & & {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle L(g)} \\ L(X') & \xrightarrow[i']{} & P' & \xleftarrow[o']{} & L(Y'). \end{array}$$

To be more descriptive we call this double category $\mathbb{O}\mathbf{pen}(\mathsf{Petri})$.

We can equivalently describe open Petri nets using decorated cospans. There is a symmetric lax monoidal pseudofunctor $F\colon (\mathsf{FinSet}, +) \to (\mathbf{Cat}, \times)$ such that for any finite set S, the category $F(S)$ has:

- objects given by Petri nets whose set of places is S,
- morphisms given by morphisms of Petri nets that are the identity on the set of places.

By Theorem 3.4 this gives a symmetric monoidal double category $F\mathbb{C}\mathbf{sp}$. Using Theorem 3.5 we can show that $F\mathbb{C}\mathbf{sp}$ is isomorphic, as a symmetric monoidal double category, to $\mathbb{O}\mathbf{pen}(\mathsf{Petri})$. However, it seems simpler to work with open Petri nets using structured

cospans. We begin by using them to define a semantics for open Petri nets. This illustrates a general method for constructing double functors between structured cospan double categories.

4.2. **The token semantics for open Petri nets.** We have described a 'token semantics' mapping Petri nets to commutative monoidal categories. Now we would like to go further and extend this to a semantics for *open* Petri nets. This should send open Petri nets to 'open commutative monoidal categories'.

To define these, we use the left adjoint functor $L' \colon \mathsf{Set} \to \mathsf{CMC}$ sending any set X to the free commutative monoidal category on this set, which has $\mathbb{N}[X]$ as its set of objects, and only identity morphisms. The category CMC is cocomplete [19, Thm. 16]. Thus, all the machinery is in place to use Theorem 3.2 to define a symmetric monoidal double category ${}_{L'}\mathbb{C}\mathbf{sp}$ where:

- an object is a set,
- a tight morphism is a function,
- a loose morphism is an **open commutative monoidal category**, that is, a cospan in CMC of the form

$$\begin{array}{ccccc} & & C & & \\ & \overset{i}{\nearrow} & & \overset{o}{\nwarrow} & \\ L'(X) & & & & L'(Y), \end{array}$$

- a 2-cell is a **map of open commutative monoidal categories**, that is, a commutative diagram in CMC of the form

$$\begin{array}{ccccc} L'(X) & \xrightarrow{i} & C & \xleftarrow{o} & L'(Y) \\ {\scriptstyle L'(f)}\downarrow & & {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle L'(g)} \\ L'(X') & \xrightarrow{i'} & C' & \xleftarrow{o'} & L'(Y'). \end{array}$$

To be more descriptive we call this double category $\mathbb{O}\mathbf{pen}(\mathsf{CMC})$.

Next, we want to parlay the operational semantics for Petri nets

$$F \colon \mathsf{Petri} \to \mathsf{CMC}$$

into an operational semantics for open Petri nets, which should be a symmetric monoidal double functor

$$\mathbb{O}\mathbf{pen}(F) \colon \mathbb{O}\mathbf{pen}(\mathsf{Petri}) \to \mathbb{O}\mathbf{pen}(\mathsf{CMC}).$$

To do this, we can use a general method for constructing double functors between structured cospan double categories:

Theorem 4.1. *Suppose we have a square in* **Cat***:*

$$\begin{array}{ccc} \mathsf{A} & \xrightarrow{L} & \mathsf{X} \\ {\scriptstyle F_0}\downarrow & \Downarrow \alpha & \downarrow{\scriptstyle F_1} \\ \mathsf{A}' & \xrightarrow{L'} & \mathsf{X}' \end{array}$$

where X *and* X' *have pushouts,* F_1 *preserves pushouts and* α *is a natural isomorphism. Then there is a double functor* $\mathbb{F} \colon {}_{L}\mathbb{C}\mathbf{sp} \to {}_{L'}\mathbb{C}\mathbf{sp}$ *such that*

- *on objects and tight morphisms,* $\mathbb{F}$ *acts as* F_0*,*
- $\mathbb{F}$ *sends any loose morphism*

$$L(a) \xrightarrow{i} x \xleftarrow{o} L(b)$$

to

$$L'(F_0(a)) \xrightarrow{F_1(i)\alpha_a} F_1(x) \xleftarrow{F_1(o)\alpha_b} L'(F_0(b))$$

- $\mathbb{F}$ *sends any 2-cell*

$$\begin{array}{ccccc} L(a) & \xrightarrow{i} & x & \xleftarrow{o} & L(b) \\ {\scriptstyle L(f)}\downarrow & & {\scriptstyle \gamma}\downarrow & & \downarrow{\scriptstyle L(g)} \\ L(a') & \xrightarrow{i'} & x' & \xleftarrow{o'} & L(b') \end{array}$$

to

$$\begin{array}{ccccc} L'(F_0(a)) & \xrightarrow{F_1(i)\alpha_a} & F_1(x) & \xleftarrow{F_1(o)\alpha_b} & L'(F_0(b)) \\ {\scriptstyle L'(F_0(f))}\downarrow & & {\scriptstyle F_1(\gamma)}\downarrow & & \downarrow{\scriptstyle L'(F_0(g))} \\ L'(F_0(a')) & \xrightarrow{F_1(i')\alpha_{a'}} & F_1(x') & \xleftarrow{F_1(o')\alpha_{b'}} & L'(F_0(b')) \end{array}$$

Proof. This is [7, Thm. 4.2]. A double functor involves extra structure besides that mentioned above, and the theorem describes that extra structure for $\mathbb{F}$. □

As usual, this result has an enhancement that covers the symmetric monoidal case. This is most easily stated using **Rex**, the 2-category of categories with finite colimits introduced in Section 3.3.

Theorem 4.2. *Suppose we have a square in* **Rex***:*

$$\begin{array}{ccc} \mathsf{A} & \xrightarrow{L} & \mathsf{X} \\ {\scriptstyle F_0}\downarrow & \Downarrow \alpha & \downarrow{\scriptstyle F_1} \\ \mathsf{A}' & \xrightarrow{L'} & \mathsf{X}' \end{array}$$

Then the double categories ${}_L\mathbb{C}\mathbf{sp}$ *and* ${}_{L'}\mathbb{C}\mathbf{sp}$ *become symmetric monoidal as in Theorem 3.2, and the double functor* $\mathbb{F}\colon {}_L\mathbb{C}\mathbf{sp} \to {}_{L'}\mathbb{C}\mathbf{sp}$ *is symmetric monoidal.*

Proof. This is [7, Thm. 4.3]. For a double functor to be symmetric monoidal is not just a property: it involves extra structure, and the theorem describes this extra structure. □

We can apply these results if we note that the free commutative monoidal category on the free Petri net on a finite set S is naturally isomorphic to the free commutative monoidal

category on S, so this square commutes up to some natural isomorphism α:

$$\begin{array}{ccc} \mathsf{FinSet} & \xrightarrow{L} & \mathsf{Petri} \\ {\scriptstyle 1}\downarrow & \Downarrow \alpha & \downarrow{\scriptstyle F} \\ \mathsf{FinSet} & \xrightarrow{L'} & \mathsf{CMC} \end{array}$$

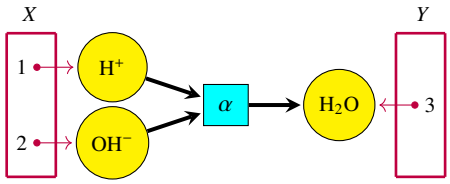
Furthermore this is a square in **Rex**. We thus obtain a symmetric monoidal double functor $\mathbb{F}\colon {}_L\mathbb{C}\mathbf{sp} \to {}_{L'}\mathbb{C}\mathbf{sp}$, which we call

$$\mathbb{O}\mathbf{pen}(F)\colon \mathbb{O}\mathbf{pen}(\mathsf{Petri}) \to \mathbb{O}\mathbf{pen}(\mathsf{CMC}).$$

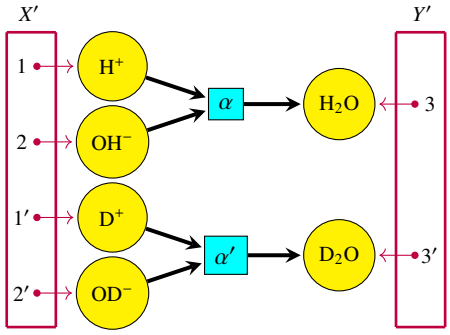
This double functor is the token semantics for *open* Petri nets.

4.3. **The uses of 2-cells.** As mentioned, one rationale for the double category approach to open systems is that the composition of these systems is associative only up to isomomorphism. But there are other advantages to using a double category. For example, the 2-cells can be used to describe maps from simple open systems to more complicated ones, and vice versa.

Here is a simple open Petri net which describes how an ionized hydrogen atom H^+ and a hydroxyl ion OH^- combine to form water:



There is an obvious 'inclusion' 2-cell from the above Petri net to this larger one:



Hydrogen has two stable isotopes, the usual one H and a heavier one called deuterium, D. In this second Petri net we are including a second reaction involving deuterium, which can create so-called 'heavy water' D_2O.

Conversely, there is a 2-cell from the second open Petri net to the first which forgets the difference between H and D, maps the transitions α and α' to α, and maps the sets X' and Y' to X and Y in the obvious way.

This example may be too simple to be interesting to chemists, but it illustrates two key uses of 2-cells in structured cospan double categories:

- We can include a simpler model of an open system in a more complicated one.

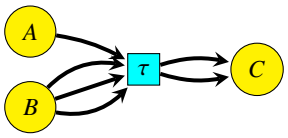
- We can project a more complicated model of an open system down to a simpler one.

Both processes are important in modeling. As we continue to change a model of an open system, either refining it or simplifying it, we should not have to treat each model we build as a separate, isolated entity. It is better, when possible, to keep track of 2-cells between them. This lets us treat the history of the modeling process as a formal entity in its own right.

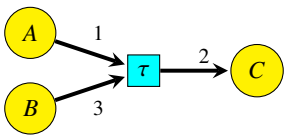
Furthermore, the category of loose morphisms and 2-cells between them may have pullbacks. This is true, for example, of $\mathbb{O}$**pen**(Petri). These pullbacks let us build more complicated open systems from simpler ones, in a process that modelers—somewhat confusingly to mathematicians—call 'stratification'. To see how pullbacks have been used to stratify Petri nets in the sphere of public health modeling, see [5].

4.4. **Whole-grain Petri nets.** There are various alternative notions of Petri net beside the one we have been using here [14]. While it is a digression, we would be remiss not to mention Kock's 'whole-grain Petri nets' [60], for two reasons. First, when people implement open Petri nets using structured cospans in category-based software, they often use whole-grain Petri nets [4, 5]. Second, while the Petri nets discussed so far present only free *commutative* monoidal categories, whole-grain Petri nets have the ability to present free strict symmetric monoidal categories.

In the Petri nets discussed so far, the arrows between places and transitions have no real individuality: permuting them has no effect. Thus, instead of drawing a *finite set* of arrows from B to τ or from τ to C in this picture:



it would be more honest to put a *natural number* on each arrow, like this:



However, in a whole-grain Petri net there really is a finite set I of arrows called **input arcs** going from places to transitions, and a finite set O of **output arcs** going from transitions to places.

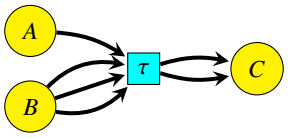
More precisely, a **whole-grain Petri net** is a diagram of finite sets

$$S \longleftarrow I \longrightarrow T \longleftarrow O \longrightarrow S.$$

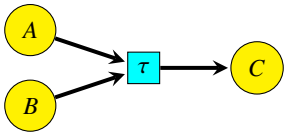
A morphism of whole-grain Petri nets, sometimes called an **etale map**, is a diagram

$$\begin{array}{ccccccccc} S & \longleftarrow & I & \longrightarrow & T & \longleftarrow & O & \longrightarrow & S \\ \downarrow & & \downarrow & \lrcorner & \downarrow & & \downarrow & & \downarrow \\ S' & \longleftarrow & I' & \longrightarrow & T' & \longleftarrow & O' & \longrightarrow & S'. \end{array}$$

Without the pullback conditions here, there would be a map from this whole-grain Petri net:



to this one:



Just as there is a left adjoint functor

$$F\colon \mathsf{Petri} \to \mathsf{CMC},$$

there is a left adjoint

$$F_{\mathsf{wg}}\colon \mathsf{wgPetri} \to \mathsf{SSMC}$$

from the category $\mathsf{wgPetri}$ of whole-grain Petri nets and etale maps to the category of strict symmetric monoidal categories and strict symmetric monoidal functors. Just as the functor F gives rise to a symmetric monoidal double functor

$$\mathbb{O}\mathbf{pen}(F)\colon \mathbb{O}\mathbf{pen}(\mathsf{Petri}) \to \mathbb{O}\mathbf{pen}(\mathsf{CMC}),$$

the functor F_{wg} gives a symmetric monoidal double functor

$$\mathbb{O}\mathbf{pen}(F_{\mathsf{wg}})\colon \mathbb{O}\mathbf{pen}(\mathsf{Petri}_{\mathsf{wg}}) \to \mathbb{O}\mathbf{pen}(\mathsf{CMC}_{\mathsf{wg}}).$$

Moreover, there is a square of symmetric monoidal double functors, commuting up to isomorphism:

$$\begin{array}{ccc} \mathbb{O}\mathbf{pen}(\mathsf{Petri}_{\mathsf{wg}}) & \xrightarrow{\mathbb{O}\mathbf{pen}(F)} & \mathbb{O}\mathbf{pen}(\mathsf{SSMC}) \\ \downarrow{\scriptstyle \mathbb{O}\mathbf{pen}(H)} & & \downarrow{\scriptstyle \mathbb{O}\mathbf{pen}(G)} \\ \mathbb{O}\mathbf{pen}(\mathsf{Petri}) & \xrightarrow{\mathbb{O}\mathbf{pen}(F_{\mathsf{wg}})} & \mathbb{O}\mathbf{pen}(\mathsf{CMC}) \end{array}$$

where $\mathbb{O}\mathbf{pen}(G)$ comes from a left adjoint functor $G\colon \mathsf{Petri}_{\mathsf{wg}} \to \mathsf{Petri}$, and $\mathbb{O}\mathbf{pen}(H)$ comes from a left adjoint $H\colon \mathsf{SSMC} \to \mathsf{CMC}$. For details, see [14, Sec. 9].

5. Dynamical systems

Next we consider a double category of open dynamical systems, defined using decorated cospans. There are many kinds of dynamical system, but we shall discuss only one, as an example: a vector field v on $\mathbb{R}^n$, which we can treat as simply a function $v\colon \mathbb{R}^n \to \mathbb{R}^n$. This determines a first-order ordinary differential equation

$$\frac{d}{dt}x(t) = v(x(t)) \tag{1}$$

which describes how a point $x(t) \in \mathbb{R}^n$ depends on time $t \in \mathbb{R}$. Higher-order ordinary differential equations can be put into first-order form by including extra variables. In physics we commonly treat $\mathbb{R}^n$ as the set of states of some system, and use a differential equation as above to describe how states evolve in time.

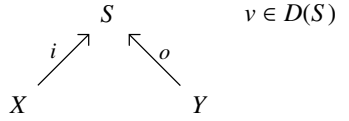
More conceptually, we can replace $\mathbb{R}^n$ with $\mathbb{R}^S$ where S is any finite set, whose elements we call **variables**. To each variable $a \in S$ we associate a real-valued function of time, say $[a]\colon \mathbb{R} \to \mathbb{R}$. The function $x\colon \mathbb{R} \to \mathbb{R}^S$ contains the information in all these functions. We can drop the brackets and use the same notation for the variables and their corresponding functions, but for expository purposes we wish to clarify the distinction between them.

To make this idea precise we should single out some class of vector fields: for example continuous, smooth, etc. There is a lot of flexibility here. If we restrict to smooth bounded vector fields, the differential equation will have a unique solution for all times for any initial data $x(0)$, and this solution $x(t)$ will be a smooth function of time. However, in our applications to Petri nets in Section 4, we need smooth vector fields that may not be bounded. For the sake of specificity, we shall use these. Beware: in this case, while the differential equation is guaranteed to have a smooth solution 'locally in time' for any initial condition $x(0)$, the solution may shoot off to infinity and become undefined after a while.

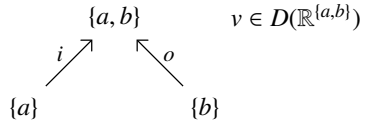
With these decisions made, for any finite set S we define

$$D(S) = \{v\colon \mathbb{R}^S \to \mathbb{R}^S \mid v \text{ is smooth}\}.$$

We define a **dynamical system** to be a finite set S together with an element of $D(S)$. We then define an **open dynamical system** to be a cospan of finite sets where the apex, say S, is decorated by an element of $D(S)$:



How can we compose open dynamical systems? Let us look at an example. Consider this open dynamical system:

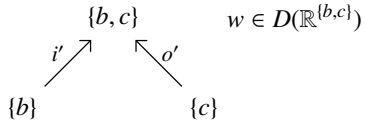


where for simplicity the functions i and o map each variable to the like-named variable. The dynamical system here specifies the differential equations

$$\begin{aligned} \frac{d[a]}{dt} &= v_a([a],[b]) \\ \frac{d[b]}{dt} &= v_b([a],[b]) \end{aligned} \tag{2}$$

where $[a], [b]\colon \mathbb{R} \to \mathbb{R}$ are the functions associated to the variable names $a, b \in X$.

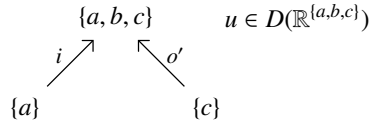
Suppose we compose the above open dynamical system with the following one:



which describes the differential equations

$$\begin{aligned} \frac{d[b]}{dt} &= w_b([b],[c]) \\ \frac{d[c]}{dt} &= w_c([b],[c]) \end{aligned} \tag{3}$$

Since we compose cospans by taking pushouts, the composite will be of the form



where u is some vector field.

What should u be? We want to combine Equations (2) and (3) somehow, identifying the variable b in the first set of equations with the like-named variable in the second set—not because they happen to have the same name, but because those variables get identified when taking the pushout. Since we have two different equations describing $d[b]/dt$, we need to reconcile these somehow. We do what a physicist would do, and *add* the right-hand sides of these equations:

$$\begin{aligned} \frac{d[a]}{dt} &= v_a([a],[b]) \\ \frac{d[b]}{dt} &= v_b([a],[b]) + w_b([b],[c]) \\ \frac{d[c]}{dt} &= w_c([b],[c]) \end{aligned} \tag{4}$$

Since $[b]$ is changing for two different reasons, we sum those effects. This is a nontrivial decision on our part, and we could decide to do something else. But this choice is surprisingly effective. For example, in classical mechanics, when an object is affected by several forces, the rate of change of its velocity is the *sum* of the rates of change you would predict from each force individually. In chemistry, when molecules of a certain sort are getting created by several reactions simultaneously, the rate at which their number changes is the *sum* of the rates of change due to the various individual reactions. We shall see examples of both kinds.

From Equation (4) we can read off the vector field u:

$$u([a],[b],[c]) = \big(v_a([a],[b]),\ v_b([a],[b]) + w_b([b],[c]),\ w_c([b],[c])\big).$$

But how do we formalize this method of composing open dynamical systems *in general*, not just in this one example? We can use theory of decorated cospans.

First, we need some generalities. Given any function $f\colon S \to S'$ between finite sets, we define the **pullback** $f^*\colon \mathbb{R}^{S'} \to \mathbb{R}^S$ to be the linear map given by

$$f^*(\psi)(s) = \psi(f(s))$$

for all $\psi \in \mathbb{R}^{S'}$, $s \in S$. This defines a contravariant functor from FinSet to $\mathsf{FinVect}_{\mathbb{R}}$, the category of finite-dimensional real vector spaces and linear maps. We define the **pushforward** $f_*\colon \mathbb{R}^S \to \mathbb{R}^{S'}$ by

$$f_*(\psi)(s') = \sum_{\{s \in S\,:\, f(s) = s'\}} \psi(s)$$

for all $\psi \in \mathbb{R}^S$, $s' \in S$. This defines a covariant functor from FinSet to $\mathsf{FinVect}_{\mathbb{R}}$.

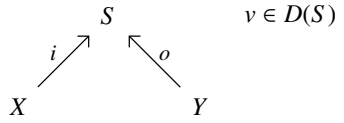
Then, we can show there is a symmetric lax monoidal functor $D\colon \mathsf{FinSet} \to \mathsf{Set}$ such that:

- D maps any finite set S to
$$D(S) = \{v\colon \mathbb{R}^S \to \mathbb{R}^S \mid v \text{ is smooth}\},$$
- D maps any function $f\colon S \to S'$ between finite sets to the function $D(f)\colon D(S) \to D(S')$ given by
$$D(f)(v) = f_* \circ v \circ f^*$$
for all $v \in D(S)$,
- the laxator $\delta_{S,S'}\colon D(S) \times D(S') \to D(S + S')$ is given by
$$\delta_{S,S'}(v, v') = i_* \circ v \circ i^* + i'_* \circ v' \circ i'^*,$$
where $i\colon S \to S + S'$ and $i'\colon S' \to S + S'$ are the inclusions into the coproduct.
- the unitor is the unique map $\delta\colon 1 \to D(\emptyset)$, since there is only one vector field on the empty set.

For details, see [20, Sec. 6]. The laxator is chosen to create the summing effect we saw in our earlier example.

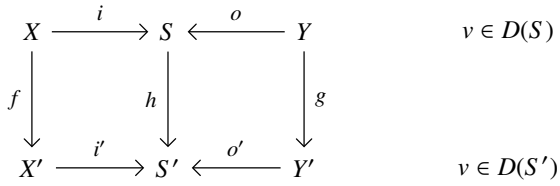
Since every set gives a discrete category with that set of objects, we can reinterpret D as a symmetric lax monoidal functor $D\colon (\mathsf{FinSet}, +) \to (\mathbf{Cat}, \times)$. Since a functor to **Cat** is a special case of a pseudofunctor, we can use Theorem 3.4 and obtain a symmetric monoidal double category $D\mathbb{C}\mathbf{sp}$ of **open dynamical systems**. In this double category

- an object is a finite set,
- a tight morphism is a function,
- a loose morphism from a finite set X to a finite set Y is an open dynamical system



$v \in D(S)$

- a 2-cell is a commuting diagram

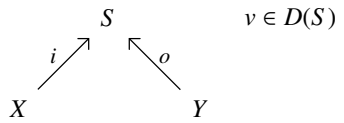


$v \in D(S)$

$v \in D(S')$

in FinSet such that $D(h)(v) = v'$.

5.1. **The open dynamical system equation.** We have seen how open dynamical systems can be composed. But we have not fully lived up to one of our claims: that we can use double categories to study systems that interact with their environment. We have seen how an open dynamical system



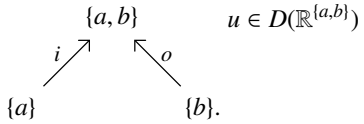
$v \in D(S)$

determines a differential equation

$$\frac{d}{dt}x(t) = v(x(t)).$$

However, this is a so-called 'autonomous' differential equation, in which the state of the system at any time completely determines its future state, without the intrusion of any influences from the outside world. So far, the open dynamical system interacts with other systems only after we compose them as loose morphisms in $D\mathbb{C}\mathbf{sp}$. We now generalize to a formalism where the outside world, *not modeled by any particular dynamical system*, can affect an open dynamical system through its interfaces.

For example, return to this open dynamical system:



Suppose we add terms to its differential equation, Equation (2), as follows:

$$\begin{aligned} \frac{d[a]}{dt} &= u_1([a],[b]) + f(t) \\ \frac{d[b]}{dt} &= u_2([a],[b]) + g(t) \end{aligned} \tag{5}$$

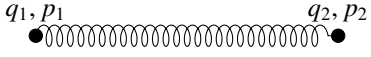
Now the equation is no longer autonomous, because f and g are arbitrary functions of time. They represent additional 'external influences'.

More generally, let $I\colon \mathbb{R} \to \mathbb{R}^X$ and $O\colon \mathbb{R} \to \mathbb{R}^Y$ be arbitrary smooth functions of time. We can compose these with the pushforward maps $i_*\colon \mathbb{R}^X \to \mathbb{R}^S$ and $o_*\colon \mathbb{R}^Y \to \mathbb{R}^S$ to add extra terms to our autonomous differential equation, Equation (1), and obtain the **open dynamical system equation**

$$\frac{dx(t)}{dt} = v(x(t)) + i_*(I(t)) - o_*(O(t)) \tag{6}$$

The minus sign is just a matter of convention. It breaks the symmetry between left and right interfaces, so it may be considered undesirable. We could leave it out altogether. However, we shall soon turn to some examples where the variables represent 'stocks' or 'concentrations'—loosely, amounts of stuff—which change in time due to 'flows'. Then it is common to adopt a convention where flows from left to right are given a positive sign, while flows in the other direction are given a negative sign. In this situation it is reasonable to call I the **inflow** and O the **outflow**.

5.2. **Classical mechanics.** Let us see this formalism at work in classical mechanics. Suppose we have two particles on the line, connected to each other by a spring. We can describe them as a dynamical system with four variables, since each particle's state is described by one position variable q_i and one momentum variable p_i:



Suppose the ith particle has mass $m_i \in \mathbb{R}$ and the spring has spring constant k. The first particle feels a force $k(q_2 - q_1)$ pulling it toward the second, while the second feels a force

$k(q_1 - q_2)$ pulling it toward the first. Since momentum is mass times velocity and force is the time derivative of momentum, we have

$$\begin{aligned} \frac{d}{dt}q_1(t) &= p_1(t)/m_1 \\ \frac{d}{dt}p_1(t) &= k(q_2(t) - q_1(t)) \\ \frac{d}{dt}q_2(t) &= p_2(t)/m_2 \\ \frac{d}{dt}p_2(t) &= k(q_1(t) - q_2(t)) \end{aligned} \tag{7}$$

Now we are acting like physicists and not distinguishing the variables q_i, p_i from the functions $[q_i]\colon \mathbb{R} \to \mathbb{R}, [p_i]\colon \mathbb{R} \to \mathbb{R}$ they name. If we let

$$x(t) = (q_1(t), p_1(t), q_2(t), p_2(t)) \in \mathbb{R}^4$$

we can write Equation (7) more tersely as

$$\frac{d}{dt}x(t) = v(x(t))$$

where

$$v(q_1, p_1, q_2, p_2) = (p_1/m_1, k(q_2 - q_1), p_2/m, k(q_1 - q_2)).$$

We can also treat this system as an *open* dynamical system, for example by decreeing that the first particle is the left interface and the second is the right interface:

$$\begin{array}{ccc} \{q_1, p_1, q_2, p_2\} & & v \in D(\mathbb{R}^{\{q_1, p_1, q_2, p_2\}}) \\ {\scriptstyle i}\nearrow & \nwarrow{\scriptstyle o} & \\ \{q_1, p_1\} & \{q_2, p_2\}. & \end{array} \tag{8}$$

What does the open dynamical system equation, Equation (6), look like in this case? For simplicity suppose we take

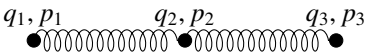
$$I(t) = (0, F_1(t)), \qquad O(t) = (0, F_2(t)).$$

Then the open dynamical system equation becomes

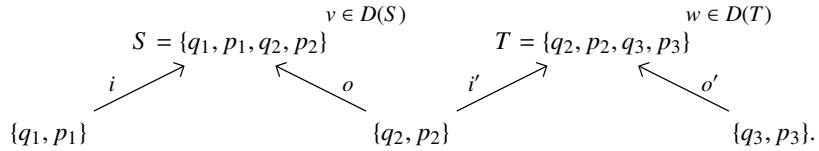
$$\begin{aligned} \frac{d}{dt}q_1 &= p_1/m_1 \\ \frac{d}{dt}p_1 &= k(q_2 - q_1) + F_1(t) \\ \frac{d}{dt}q_2 &= p_2/m_2 \\ \frac{d}{dt}p_2 &= k(q_1 - q_2) - F_2(t). \end{aligned}$$

This system of equations describes two rocks connected by a spring where the first is pushed to the right by an external force $F_1(t)$ and the second is pushed to the left by an external force $F_2(t)$. If the sign convention here feels unnatural, the reader is free to get rid of the minus sign in Equation (6), but we will try to justify the sign in the next two sections.

Exercise 5.1. *Describe a system of three particles connected by springs:*



as an open dynamical system. To do this, compute the composite of two open dynamical systems of the kind just described, regarded as loose morphisms in $D\mathbb{C}\mathbf{sp}$*:*



Exercise 5.2. *Figure out the open dynamical system equation for the system in Exercise 5.1, assuming the first particle feels an external force* $F_1(t)$ *pushing to the left, while the third feels an external force* $F_3(t)$ *pushing to the right. In this example the second particle is not part of an interface, so it feels no external force.*

The reader may wonder whether such an elaborate formalism is required to study three rocks connected by two springs. It is not. What the formalism does is to make explicit the physicist's intuitive understanding of how to take two systems of differential equations, each describing a physical system, and combine them to describe a larger system built from those two parts. This should be useful both for mathematicians wishing to prove theorems about composites of open systems, and also people who want to automate the process of modeling composite systems.

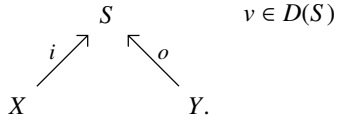
The examples just discussed can be vastly generalized. For systems of finitely many particles moving in $\mathbb{R}^n$ and interacting by arbitrary forces, this is straightforward. It is more challenging to generalize the framework to handle open Lagrangian systems where the configuration space is a manifold, open Hamiltonian systems where the phase space is a symplectic manifold , and port-Hamiltonian systems where the phase space is a Dirac manifold. Much work has been done on these issues [11, 22, 30, 31, 65], but many interesting open questions remain.

5.3. **Black-boxing.** Next let us take an open dynamical system and extract from it the relations between externally observable quantities that holds whenever the system is in a 'steady state': a state where nothing is changing. The process of extracting this relation is an example of what we call 'black-boxing', since it discards information that cannot be seen at the interfaces. We shall see that it defines a double functor

$$\blacksquare \colon D\mathbb{C}\mathbf{sp} \to \mathbb{R}\mathbf{el}$$

from the double category of open dynamical systems to the double category of relations. The functoriality of black-boxing implies that we can compose two open dynamical systems and then black-box them, or black-box each one and compose the resulting relations: either way, the final answer is the same.

Consider an open dynamical system $F \colon X \nrightarrow Y$ given by the decorated cospan



If we take its open dynamical system equation, Equation (6), and fix x, I and O to be constant in time, we can treat them as vectors $x \in \mathbb{R}^S$, $I \in \mathbb{R}^X$, $O \in \mathbb{R}^Y$. Then the equation reduces to

$$v(x) + i_*(I) - o_*(O) = 0.$$

Thus, we define a **steady state** with inflows $I \in \mathbb{R}^X$ and outflows $O \in \mathbb{R}^Y$ to be a vector $x \in \mathbb{R}^S$ for which the above equation holds. We define the **black-boxing** of our open dynamical system to be the set

$$\blacksquare(F) \subseteq \mathbb{R}^X \times \mathbb{R}^X \times \mathbb{R}^Y \times \mathbb{R}^Y$$

consisting of all 4-tuples $(i^*(x), I, o^*(x), O)$ where $x \in \mathbb{R}^S$ is a steady state with inflows $I \in \mathbb{R}^X$ and outflows $O \in \mathbb{R}^Y$:

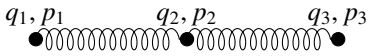
$$\blacksquare(F) = \left\{ (i^*(x), I, o^*(x), O) \,\middle|\, x \in \mathbb{R}^S,\ I \in \mathbb{R}^X,\ O \in \mathbb{R}^Y,\ v(x) + i_*(I) - o_*(O) = 0 \right\}. \quad (9)$$

The idea is that black-boxing an open dynamical system records its 'externally observable steady state behavior'.

In the example from Section 5.2, with two particles connected by a spring, we get

$$\blacksquare(F) = \left\{ \Big((q_1, 0), (0, k(q_1 - q_2)), (q_2, 0), (0, k(q_1 - q_2)) \Big) \,\middle|\, q_1, q_2 \in \mathbb{R} \right\}.$$

This says that in steady state, the particles can be at rest in any position, with equal and opposite forces acting on them, chosen precisely to counteract the force of the spring. Of course this example is somewhat degenerate because every variable for the system is 'exposed', that is, also a variable for an interface. A more typical example would involve three particles connected by springs, with the first particle being the left interface and the second being the right interface:



Exercise 5.3. *Black-box the composite system of Exercise 5.1, using the open dynamical system equation obtained in Exercise 5.2.*

Now let us return to the general theory. Given an open dynamical system F with left interface X and right interface Y, we can think of $\blacksquare(F)$ as a relation from $\mathbb{R}^X \times \mathbb{R}^X$ to $\mathbb{R}^Y \times \mathbb{R}^Y$. There is a double category $\mathbb{R}\mathbf{el}$ where:

- an object is a set,
- a tight morphism from X to Y is a function $f \colon X \to Y$,
- a loose morphism from X to Y is a relation $R \colon X \nrightarrow Y$, meaning a subset $R \subseteq X \times Y$,
- a 2-cell is a square

$$\begin{array}{ccc} X_1 & \overset{R_1}{\longmapsto} & Y_1 \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ X_2 & \underset{R_2}{\longmapsto} & Y_2 \end{array}$$

  obeying $(f \times g)R_1 \subseteq R_2$,
- composition of tight morphisms is composition of functions,
- composition of loose morphisms is the usual composition of relations,
- vertical and horizontal composition of 2-cells is done in the only way possible.

The last item deserves some explanation. We say a double category is **degenerate** if given any **frame**—that is, any collection of objects, vertical 1-morphisms and horizontal 1-cells as follows:

$$\begin{array}{ccc} X_1 & \xrightarrow{M} & Y_1 \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ X_2 & \xrightarrow{N} & Y_2 \end{array}$$

there exists at most one 2-cell

$$\begin{array}{ccc} X_1 & \xrightarrow{M} & Y_1 \\ {\scriptstyle f}\downarrow & \Downarrow \alpha & \downarrow{\scriptstyle g} \\ X_2 & \xrightarrow{N} & Y_2 \end{array}$$

filling this frame. In a degenerate double category, like $\mathbb{R}\mathbf{el}$, there is no choice in how to define the vertical or horizontal composite of 2-cells once composition of loose and tight morphisms is fixed. Thus we do not need to check the laws governing composition of 2-cells: we need merely check that the composites exist, which is easy to do in the case of $\mathbb{R}\mathbf{el}$.

In this language, black-boxing maps any loose morphism in $D\mathbb{C}\mathbf{sp}$, namely an open dynamical system $F\colon X \nrightarrow Y$, to a loose morphism in $\mathbb{R}\mathbf{el}$, namely

$$\blacksquare(F)\colon \mathbb{R}^X \times \mathbb{R}^X \nrightarrow \mathbb{R}^Y \times \mathbb{R}^Y.$$

This immediately leads to the question of whether black-boxing extends to a functor from $D\mathbb{C}\mathbf{sp}$ to $\mathbb{R}\mathbf{el}$. We shall see it does.

Furthermore, the double category $\mathbb{R}\mathbf{el}$ is symmetric monoidal in a natural way, where the tensor product of objects and tight morphisms is given by the cartesian product in Set, while the tensor product of loose morphisms (i.e. relations) is given by the cartesian product of their underlying subsets. We can now state the main theorem about black-boxing:

Theorem 5.4. *There is a symmetric monoidal double functor $\blacksquare\colon D\mathbb{C}\mathbf{sp} \to \mathbb{R}\mathbf{el}$ sending*

- *any finite set X to the vector space $\mathbb{R}^X \times \mathbb{R}^X$,*
- *any function $f\colon X \to Y$ to the pushforward $f_*\colon \mathbb{R}^X \times \mathbb{R}^X \to \mathbb{R}^Y \times \mathbb{R}^Y$,*
- *any open dynamical system $F\colon X \nrightarrow Y$ to its black-boxing $\blacksquare(F)$ as defined in Equation (9),*
- *any 2-cell to the only 2-cell with the appropriate frame.*

Proof. A closely related theorem was proved at the level of categories in [20, Thm. 23]. It was shown that isomorphic open dynamical systems $F\colon X \nrightarrow Y$ have the same black-boxing $\blacksquare(F)$, and there is a symmetric monoidal functor from $D\mathsf{Csp}$ to Rel sending any isomorphism class of open dynamical systems $F\colon X \nrightarrow Y$ to $\blacksquare(F)$. (In fact it was shown that this functor maps F to a relation of a special kind, called a 'semialgebraic' relation, but this is irrelevant here.) At the level of double categories, this theorem implies that $\blacksquare$ preserves composition of horizontal morphisms on the nose. Clearly it preserves composition of vertical morphisms. Since $\mathbb{R}\mathbf{el}$ is a degenerate double category there is at most one choice of where $\blacksquare$ can send any 2-cell, and the symmetric monoidality of the double functor $\blacksquare$ follows easily from the corresponding result at the level of categories. □

The functoriality is the most interesting step in proving the above theorem. One needs to check that when two open dynamical systems are composed, one can compose their steady states to form a steady state of the composite system when the outflow of the first steady state equals the inflow of the second.

5.4. **Structured versus decorated cospans, revisited.** Before moving on, let us briefly return to a technical issue raised in Section 3.3. We used decorated cospans to define a double category of open dynamical systems, $D\mathbb{C}\mathbf{sp}$. Can we define an equivalent double category using structured cospans?

It seems not, but at present the main evidence comes from [9, Thm. 4.1]. This result says we *could* define such a structured cospan double category if the natural lift of $D\colon \mathsf{FinSet} \to \mathbf{Cat}$ to a pseudofunctor $D\colon \mathsf{FinSet} \to \mathbf{SymMonCat}$ factored through **Rex**. It also says that in this case, the opfibration $U\colon \int D \to \mathsf{FinSet}$ would be a right adjoint. However, U is not a right adjoint. After all, this would imply that for every $S \in \mathsf{FinSet}$ the comma category $S \downarrow U$ has an initial object. But this is not true. Because the empty set is initial in FinSet, the comma category $\emptyset \downarrow U$ is just $\int D$. This contains an object $(\emptyset, v_\emptyset)$, where $v_\emptyset$ is the only possible vector field on $\mathbb{R}^\emptyset$, namely, the zero vector field. The only object in $\int D$ with any morphisms to $(\emptyset, v_\emptyset)$ is $(\emptyset, v_\emptyset)$ itself, so no other object can be initial. However $(\emptyset, v_\emptyset)$ is not initial either, because it has no morphisms to an object (S, v) unless v is the zero vector field on $\mathbb{R}^S$.

While this argument is impressively technical, its conclusions are quite weak. It does not imply that no structured cospan double category is equivalent to $D\mathbb{C}\mathbf{sp}$. It merely says that one particular strategy for constructing such a structured cospan double category fails.

Challenge 5.5. *Find conditions implying that a decorated cospan double category is* not *equivalent to any structured cospan double category.*

Luckily, there is a practical workaround in this case: a structured cospan category of 'linearly parametrized' dynamical systems [1, Sec. 3.1]. Here instead of equipping a finite set S with a fixed vector field $v \in D(S)$, we equip it with a parametrized family of vector fields—or more precisely a finite set P and a linear map $v\colon \mathbb{R}^P \to D(S)$. We call S the set of **variables** and P the set of **parameters**. This captures the fact that in physics and other subjects, the vector fields describing time evolution often depend linearly on some P-tuple of real numbers called 'parameters' or 'coupling constants'.

There is a category $\mathsf{ParaDynam}$ for which:

- an object is a **linearly parametrized dynamical system**: a pair of finite sets S and P and a linear map $v\colon \mathbb{R}^P \to D(S)$;
- a morphism from (S, P, v) to (S', P', v') is a pair of functions $f\colon S \to S', q\colon P \to P'$ making this square commute:

$$\begin{array}{ccc} \mathbb{R}^P & \xrightarrow{v} & D(S) \\ {\scriptstyle q_*}\downarrow & & \downarrow{\scriptstyle f_* \circ - \circ f^*} \\ \mathbb{R}^{P'} & \xrightarrow{v'} & D(S') \end{array}$$

The category $\mathsf{ParaDynam}$ has finite colimits [1, Prop. 3.5], and the forgetful functor $U\colon \mathsf{ParaDynam} \to \mathsf{FinSet}$ sending any linearly parametrized dynamical system to its set of variables has a left adjoint L sending any finite set to the linearly parametrized dynamical system with the empty set of parameters. Thus, we can construct a structured cospan

double category of open linearly parametrized dynamical systems and show that it is symmetric monoidal [1, Prop. 3.9].

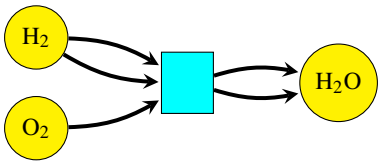
6. Petri nets with rates

Even if we fix a class of dynamical systems, there are typically many choices of syntax for describing these systems. This is certainly true of the dynamical systems we have just discussed: systems of first-order ordinary differential equations. We explain one syntax here, namely 'Petri nets with rates'. These are used in chemistry, population biology, epidemiology and other fields [5, 35, 51, 58, 88]. Here we describe a double category of *open* Petri nets with rates, and a double functor from this to our double category of open dynamical systems, $D\mathbb{C}\mathbf{sp}$. The easiest way to understand all this starts with chemistry.

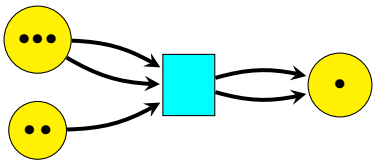
6.1. **Chemistry.** Chemists sometimes use Petri nets where the places represent 'chemical species': different kinds of molecules, elements and ions. Then transitions represent chemical reactions. For example, we can describe the formation of water from hydrogen and oxygen molecules

$$2H_2 + O_2 \longrightarrow 2H_2O$$

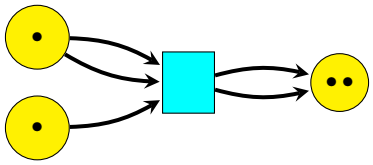
using this Petri net P:



The transition in aqua here describes a reaction where two molecules of hydrogen and one of oxygen become two molecules of water. We could use the token semantics described in Section 4 to model how individual molecules react in this way. For example, there is a morphism in the category FP from the marking



to the marking



However, chemists often deal with vast numbers of molecules in solution. Instead of counting molecules individually they count them in 'moles': a mole is about $6 \cdot 10^{23}$ molecules. Then the 'concentration' of a given species, measured in moles per liter, is treated approximately as a smoothly varying real-valued function of time, and we want differential equations describing how these concentrations change. For this chemists commonly use a recipe called the 'law of mass action'.

It is easiest to explain this with an example. For the Petri net above, the law of mass action gives these differential equations:

$$\begin{aligned} \frac{d}{dt}[\mathrm{H}_2] &= -2r_1\,[\mathrm{H}_2]^2\,[\mathrm{O}_2] \\ \frac{d}{dt}[\mathrm{O}_2] &= -r_2\,[\mathrm{H}_2]^2\,[\mathrm{O}_2] \\ \frac{d}{dt}[\mathrm{H}_2\mathrm{O}] &= 2r_1\,[\mathrm{H}_2]^2\,[\mathrm{O}_2]. \end{aligned}$$

Here $[\mathrm{H}_2] \colon \mathbb{R} \to \mathbb{R}$ is the concentration of hydrogen molecules as a function of time, and similarly for $[\mathrm{O}_2]$ and $[\mathrm{H}_2\mathrm{O}]$. The constant $r_1 \in [0, \infty)$ is the 'rate constant' of this particular reaction, which depends on various environmental conditions. All the time derivatives are proportional to $[\mathrm{H}_2]^2\,[\mathrm{O}_2]$, because this chemical reaction takes two hydrogen molecules and one oxygen molecule as inputs. We should imagine molecules randomly moving around; then the probability that two H_2 molecules and one O_2 molecules are in the same very small region of space is approximately proportional to $[\mathrm{H}_2]^2\,[\mathrm{O}_2]$. The time derivative of $[\mathrm{H}_2]$ is also proportional to -2, because 2 hydrogen molecules get destroyed in this reaction. Similarly, $\frac{d}{dt}[\mathrm{O}_2]$ is proportional to -1 because 1 oxygen molecule gets destroyed, and $\frac{d}{dt}[\mathrm{H}_2\mathrm{O}]$ is proportional to 1 because one water molecule gets created.

Generalizing from this example, we can state the law of mass action precisely as follows. We start with a **Petri net with rates**, meaning a Petri net $s, t \colon T \to \mathbb{N}[S]$ together with a function $r \colon T \to [0, \infty)$ assigning to each transition $\tau \in T$ a nonnegative real number $r(\tau)$ called its **rate constant**. In chemistry the elements of S are called **species** rather than places.

From our Petri net with rates, we get a differential equation

$$\frac{d}{dt} x(t) = v(x(t))$$

called the **rate equation** describing the evolution of a function $x \colon \mathbb{R} \to \mathbb{R}^S$. This function specifies the concentration of each species as a function of time. The vector field v on $\mathbb{R}^S$ is given by **law of mass action**:

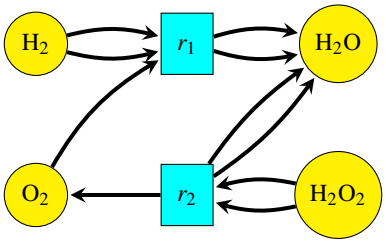
$$v(x) = \sum_{\tau \in T} r(\tau)\,(t(\tau) - s(\tau)) x^{s(\tau)} \tag{10}$$

where $x \in \mathbb{R}^S$. The notation here requires some explanation. We think of $t(\tau), s(\tau) \in \mathbb{N}[S]$ as vectors in $\mathbb{R}^S$. Their difference $t(\tau) - s(\tau)$ says, for each species, the number of tokens of that species created by the transition τ minus the number destroyed. Finally, we define

$$x^{s(\tau)} = \prod_{i \in S} {x_i}^{s(\tau)_i}.$$

Here for any $i \in S$ we write $x_i \in \mathbb{R}$ for the ith component of $x \in \mathbb{R}^S$, i.e., the concentration of the ith species. Similarly, we write $s(\tau)_i$ for the ith component of $s(\tau) \in \mathbb{R}^S$.

Exercise 6.1. *Here is a Petri net with rates:*



Transitions are labeled with their rate constants. The transition with rate constant r_1 represents the oxidation of hydrogen to form water as before, while the transition with rate constant r_2 represents the decomposition of hydrogen peroxide. Use the law of mass action to derive the following differential equations:

$$\begin{aligned}
\frac{d}{dt}[\mathrm{H_2}] &= -2r_1\,[\mathrm{H_2}]^2\,[\mathrm{O_2}] \\
\frac{d}{dt}[\mathrm{O_2}] &= -r_1\,[\mathrm{H_2}]^2\,[\mathrm{O_2}] + r_2\,[\mathrm{H_2O_2}]^2 \\
\frac{d}{dt}[\mathrm{H_2O_2}] &= -2r_2\,[\mathrm{H_2O_2}]^2 \\
\frac{d}{dt}[\mathrm{H_2O}] &= 2r_1\,[\mathrm{H_2}]^2\,[\mathrm{O_2}] + 2r_2\,[\mathrm{H_2O_2}]^2.
\end{aligned}$$

6.2. **Open Petri nets with rates.** Now we consider *open* Petri nets with rates, and explain a semantics mapping them to open dynamical systems. The first step is to define a category of Petri nets with rates. Then we use decorated cospans to construct a double category of open Petri nets with rates. Then we use a general recipe for constructing maps between decorated cospan categories. Readers uninterested in Petri nets with rates may still be interested in this general recipe, Theorem 6.2, since it is the decorated cospan analogue of Theorems 4.1 and 4.2 for structured cospans.

The first step is simple enough. There is a category whose objects are Petri nets with rates, where a morphism from

$$[0,\infty) \xleftarrow{r} T \underset{t}{\overset{s}{\rightrightarrows}} \mathbb{N}[S]$$

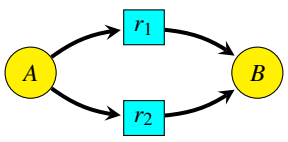
to

$$[0,\infty) \xleftarrow{r'} T' \underset{t'}{\overset{s'}{\rightrightarrows}} \mathbb{N}[S']$$

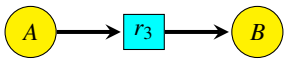
is a morphism of the underlying Petri nets whose map $g\colon T \to T'$ obeys

$$r'(\tau') = \sum_{\{\tau \in T : g(\tau) = \tau'\}} r(\tau)$$

for all $\tau' \in T'$. For example, if $S = \{A, B\}$ there is a morphism in $F(S)$ from this Petri net with rates:



to this one:



if and only if $r_3 = r_1 + r_2$.

Next, to construct a double category of open Petri nets with rates, we note that there is a symmetric lax monoidal pseudofunctor

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

such that for any finite set S:

- an object of $F(S)$ is a Petri nets with rates whose set of places is S,
- a morphism is a morphism of Petri nets with rates that is the identity on the set of places.

By Theorem 3.4, F gives a symmetric monoidal double category $F\mathbb{C}\mathbf{sp}$, which we call the double category of **open Petri nets with rates**. In this double category

- an object is a finite set,
- a tight morphism is a function,
- a loose morphism from a finite set X to a finite set Y is an **open Petri net with rates**, meaning a cospan of finite sets

  $$X \xrightarrow{i} S \xleftarrow{o} Y$$

  decorated with a Petri net with rates

  $$[0,\infty) \xleftarrow{r} T \underset{t}{\overset{s}{\rightrightarrows}} \mathbb{N}[S],$$

- a 2-cell is a map of cospans of finite sets

  $$\begin{array}{ccccc} X & \xrightarrow{i} & S & \xleftarrow{o} & Y \\ {\scriptstyle f}\downarrow & & {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle g} \\ X' & \xrightarrow[i']{} & S' & \xleftarrow[o']{} & Y'. \end{array}$$

  together with a morphism of decorations, namely a morphism of Petri nets with rates from

  $$[0,\infty) \xleftarrow{r} T \underset{t}{\overset{s}{\rightrightarrows}} \mathbb{N}[S]$$

  to

  $$[0,\infty) \xleftarrow{r'} T' \underset{t'}{\overset{s'}{\rightrightarrows}} \mathbb{N}[S']$$

  where the map from S to S' is α.

Now we are ready to define a semantics mapping open Petri nets to open dynamical systems. Recall that in Section 5 we used decorated cospans to construct a symmetric monoidal double category $D\mathbb{C}\mathbf{sp}$ of open dynamical systems. This was built using a symmetric lax monoidal pseudofunctor

$$D\colon (\mathsf{FinSet}, +) \to (\mathbf{Cat}, \times)$$

that maps any finite set S to the discrete category on the set

$$\left\{v\colon \mathbb{R}^S \to \mathbb{R}^S \mid v \text{ is smooth}\right\}.$$

Our desired semantics should be a symmetric monoidal double functor

$$\blacksquare \colon F\mathbb{C}\mathbf{sp} \to D\mathbb{C}\mathbf{sp}$$

sending any open Petri net with rates to an open dynamical system. This was already defined at the level of categories in [20, Sec. 7], where it was called 'gray-boxing', since it obscures some but not all of the details of an open Petri net. To boost this result to the double category level, we use the following general recipe for defining maps between decorated cospan double categories.

Theorem 6.2. *Given categories* A *and* A' *with finite colimits, lax monoidal pseudofunctors* $F \colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ *and* $F' \colon (\mathsf{A}', +) \to (\mathbf{Cat}, \times)$*, a finite colimit preserving functor* $H \colon \mathsf{A} \to \mathsf{A}'$*, a lax monoidal pseudofunctor* $E \colon (\mathbf{Cat}, \times) \to (\mathbf{Cat}, \times)$ *and a monoidal natural transformation* θ *as in the following diagram:*

$$\begin{array}{ccc} \mathsf{A} & \xrightarrow{F} & \mathbf{Cat} \\ {\scriptstyle H}\downarrow & \Downarrow\theta & \downarrow{\scriptstyle E} \\ \mathsf{A}' & \xrightarrow[F']{} & \mathbf{Cat} \end{array}$$

we obtain a double functor $\Theta \colon F\mathbb{C}\mathbf{sp} \to F'\mathbb{C}\mathbf{sp}$*. If* F, F' *and* E *are symmetric, then* $\Theta \colon F\mathbb{C}\mathbf{sp} \to F'\mathbb{C}\mathbf{sp}$ *is a symmetric monoidal double functor.*

Proof. This is [9, Thm. 2.5], which provides the details of how Θ is defined. □

Let us take the square in this theorem to be

$$\begin{array}{ccc} \mathsf{FinSet} & \xrightarrow{F} & \mathbf{Cat} \\ {\scriptstyle 1}\downarrow & \Downarrow\theta & \downarrow{\scriptstyle 1} \\ \mathsf{FinSet} & \xrightarrow[D]{} & \mathbf{Cat}. \end{array}$$

Here θ is a monoidal natural transformation given as follows. For any finite set S, $\theta_S : F(S) \to D(S)$ maps any Petri net with rates

$$[0, \infty) \xleftarrow{r} T \underset{t}{\overset{s}{\rightrightarrows}} \mathbb{N}[S]$$

to a smooth vector field on $\mathbb{R}^S$, say v. This vector field is defined using the law of mass action, as explained in Section 6.1. Namely, for any $x \in \mathbb{R}^S$, we set

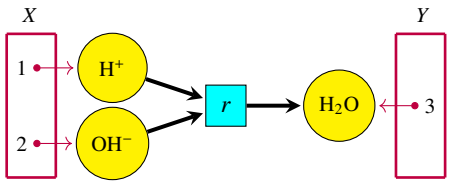
$$v(x) = \sum_{\tau \in T} r(\tau)\,(t(\tau) - s(\tau)) x^{s(\tau)}$$

where

$$x^{s(\tau)} = \prod_{i \in S} {c_i}^{s(\tau)_i}$$

and we think of $t(\tau), s(\tau) \in \mathbb{N}[S]$ as vectors in $\mathbb{R}^S$. That θ is monoidal follows from [20, Thm. 18]. Thus, it defines a symmetric monoidal double functor $\blacksquare \colon F\mathbb{C}\mathbf{sp} \to D\mathbb{C}\mathbf{sp}$.

One of the simplest consequences of this result is that we can get an open dynamical system equation from an open Petri net with rates, following the ideas in Section 5.1. For example, consider this open Petri net with rates:



gives this open dynamical system equation

$$\begin{aligned} \frac{d}{dt}[\mathrm{H}^+] &= -r\,[\mathrm{H}^+]\,[\mathrm{OH}^-] + I_1(t) \\ \frac{d}{dt}[\mathrm{OH}^-] &= -r\,[\mathrm{H}^+]\,[\mathrm{OH}^-] + I_2(t) \\ \frac{d}{dt}[\mathrm{H_2O}] &= r\,[\mathrm{H}^+]\,[\mathrm{OH}^-] - O_3(t). \end{aligned}$$

6.3. **Applications.** One might hope that open Petri nets with rates would help mathematical chemists prove new theorems. So far, things have worked out quite differently. Their main application so far has been to software for epidemiology!

Shortly after the formulation of structured cospans, the world was hit with a pandemic. At this time Fairbanks and Patterson were developing Catlab [26], a framework for applied and computational category theory, written in the Julia language. Fairbanks had already implemented decorated cospans in Catlab—but in 2020, Patterson wrote the first version of structured cospans in this framework, specifically so that Fairbanks and Halter could implement open (whole-grain) Petri nets with rates. Soon the team applied these to recreate part of the UK's main COVID model [75]. There is by now a well-developed software package to build and manipulate open Petri nets in Catlab, called AlgebraicPetri [4]. This works together with a package for open dynamical systems developed by Libkind, called AlgebraicDynamics [3]. AlgebraicDynamics is able to use either the variable sharing paradigm or the input-output paradigm.

All this work attracted the attention of Osgood and Li, computational epidemiologists who helped run COVID modeling in Canada. These experts pointed out that in public health modeling, 'stock and flow models' are more widely used than Petri nets with rates. They are similar to Petri nets with rates, but more general in some ways and less so in others. On the one hand, stock and flow models only allow transitions with either:

- one species as input and one as output (e.g. the transition of a patient from one state to another),
- one species as input and none as output (e.g. death), or
- no species as input and one as output (e.g birth).

On the other hand, they can describe dynamics more general than given by the law of mass action, and this is crucial in epidemiology.

Working with the Catlab team, Osgood and Li developed a software package called StockFlow to handle open stock and flow models [17, 86]. Osgood's student Redekopp also developed a user-friendly web-based front end for StockFlow, called ModelCollab [16, 68]. For expository papers explaining all this software in more detail, see [16, 62].

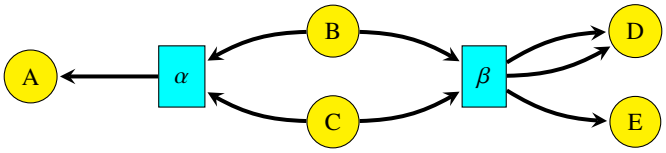
This software has the following advantages over traditional software:

- **Compositionality.** Rather than merely a piece of code, each model is a crisply defined mathematical structure designed from the start to be easily combined with other models: for example, a structured or decorated cospan. This lets models of specific subsystems be constructed individually by different domain experts, and then composed to form larger models, supporting ongoing collaboration between these parties.
- **Functorial semantics.** There is a clear distinction between a model and a specific way of extracting information from this model. This is achieved by treating different ways of extracting information from models as different functors.
- **Ease of stratification.** One can 'stratify' models—that is, create more detailed models by subdividing stocks in an existing model—without rewriting the whole model from scratch by hand. This is achieved using pullbacks.

So far, the software mentioned above does not make explict the *double* categorical nature of structured or decorated cospans, though it is implicit. Patterson is working on a new software platform, CatColab [25], based on 'double categorical doctrines'. However, this use of double categories is different from working with double categories of open systems. On the other hand, the UK agency ARIA is running a project for safeguarded AI which may develop software using Libkind and Myers' work on double categories of open systems [63, 64, 71]. We say a bit about their work in Section 7.3.

Eventually people may prove theorems about the dynamics of open Petri nets with rates, but this has not happened yet. One reason is that few have tried. Another is that while there are many interesting theorems in mathematical chemisty [39, 40, 55], and challenging problems such as the Global Attractor Conjecture [32], the Persistence Conjecture and the Permanence Conjecture [34], few involve Petri nets. Far more involve 'reaction networks'.

Petri nets and reaction networks are often considered equivalent formalisms. For example, this Petri net:



corresponds to this reaction network:

$$A \longleftarrow \mathrm{B} + \mathrm{C} \longrightarrow 2\mathrm{D} + \mathrm{E}.$$

The reaction network is a graph where each edge corresponds to a transition of the Petri net, while each vertex corresponds to a so-called 'complex': a sum of species that is the source or target of some transition. However, we compose open Petri nets by identifying species, while individual species are not a visible part of a reaction network. This leaves us with two challenges:

Challenge 6.3. *Prove theorems about the qualitative dynamics of open Petri nets with rates, perhaps building on existing results about Petri nets with rates, for example as in* [33, 35]*, where they are called 'directed species-reaction graphs'.*

Challenge 6.4. *Develop a framework for open reaction networks suitable for generalizing existing results on reaction networks and invariants such as the 'deficiency' of a reaction network* [39, 40].

7. The variable sharing paradigm

As mentioned in Section 1, there are several paradigms for open systems. Here we are solely focused on the 'variable sharing paradigm', where we compose open systems by gluing them together, or identifying variables. While we have tried to explain how structured and decorated cospans let us work within this paradigm, we have not yet analyzed which features are distinctive of this paradigm. We turn to this question now.

When he first discovered decorated cospans, Fong was studying electrical circuits [12, 42, 43]. He noticed that electrical circuits can be composed in very flexible ways. A circuit can have any number of wires coming out of it, with no fundamental distinction between inputs and outputs. We can compose circuits by arbitrarily connecting their wires. Furthermore:

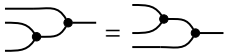
- We can take two wires coming out of a circuit and join them, getting a new circuit with one fewer wire coming out.

- We can add an extra wire to a circuit that doesn't actually connect with anything, getting a new circuit with one more wire coming out.

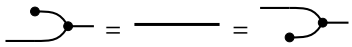
- We can take any wire coming out of a circuit and split it, getting a new circuit with one more wire coming out.

- We can take a wire coming out of a circuit and cap it off, getting a new circuit with one fewer wire coming out.

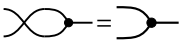
These constructions obey all the rules of a 'special commutative Frobenius monoid', meaning the associative law:



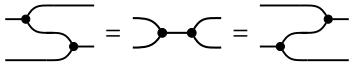
the left and right unit laws:



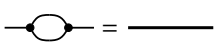
the commutative law:



where $\times$ denotes two wires crossing over each other, together with the 'Frobenius equations':



and finally the 'special' law:



By composing some of these operations we can create a 'cup' and a 'cap':

These allow us to bend wires around. Moreover, thanks to the Frobenius monoid laws, the cap and cup obey the so-called 'zigzag identities':

Fong noticed that all these are general features of decorated cospan categories. Every object in a decorated cospan category is a special commutative Frobenius monoid internal to that category, in a canonical way. Furthermore, all these Frobenius structures fit together to make the category into a 'hypergraph category'—a concept which, while apparently quite complicated, had already proved its importance by independently being discovered by multiple authors in different contexts [24, 46, 69, 57, 79].

Definition 7.1. *A* **hypergraph category** *is a symmetric monoidal category* $(\mathsf{C}, \otimes)$ *where each object* $a \in \mathsf{C}$ *is equipped with a* **multiplication** $\mu_a \colon a \otimes a \to a$, **unit** $\eta_a \colon I \to a$, **comultiplication** $\delta_a \colon a \to a \otimes a$, *and* **counit** $\epsilon_a \colon a \to I$, *obeying the laws of a special commutative Frobenius monoid and satisfying*

$$\begin{aligned} \mu_{a\otimes b} &= (\mu_a \otimes \mu_b)(1_a \otimes \sigma_{b,a} \otimes 1_b) \\ \eta_{a\otimes b} &= \eta_a \otimes \eta_b \\ \delta_{a\otimes b} &= (1 \otimes \sigma_{b,a} \otimes 1)(\delta_a \otimes \delta_b) \\ \epsilon_{a\otimes b} &= \epsilon_a \otimes \epsilon_b \end{aligned}$$

where $\sigma_{a,b} \colon a \otimes b \to b \otimes a$ *is the symmetry, as well as*

$$\eta_I = 1_I = \epsilon_I$$

where I *is the unit for the tensor product.*

Later, Fong and Spivak gave a slick equivalent definition of hypergraph category using operads built from categories of cospans [45]. However, since they were working with categories rather than double categories, we can only apply their work to a double category $\mathbb{D}$ by forming the category D whose morphisms are *isomorphism classes* of loose morphisms in $\mathbb{D}$. To get around this, we shall try to recapitulate some of their work using double categories. In the end this suggests two possible concepts of 'hypergraph double category': one that categorifies their definition, and another that categorifies Definition 7.1.

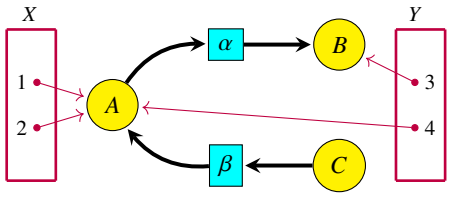
More precisely, in Section 7.1 we show that any symmetric monoidal structured or decorated cospan double category $\mathbb{D}$ has a kind of 'exoskeleton' where the loose morphisms are *trivially* structured or decorated cospans. These can be seen as cospans in $\mathbb{D}_0$, the category of objects and tight morphisms of $\mathbb{D}$. In Section 7.2, we show that this exoskeleton $\mathbb{C}\mathbf{sp}(\mathbb{D}_0)$ in turn has an 'outer shell' where the loose morphisms are cospans built up using only the objects of $\mathbb{D}_0$ and the operations present in any category with finite colimits. More precisely this outer shell is $\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0)$, where $\tilde{\mathbb{D}}_0$ is the free category with finite colimits on the objects of $\mathbb{D}$. We show there are symmetric monoidal double functors

$$\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0) \xrightarrow{j} \mathbb{C}\mathbf{sp}(\mathbb{D}) \xrightarrow{\iota} \mathbb{D}.$$

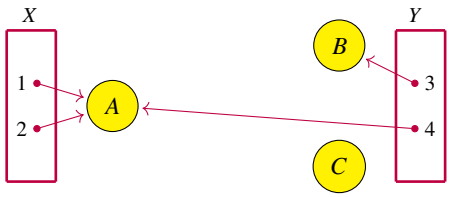
In Section 7.3 we use the outer shell to motivate Libkind and Myers' definition of hypergraph double category [64, Ex. 8.11], which categorifies Fock and Spivak's slick definition of hypergraph category. In Section 7.4 we show that any object in a symmetric monoidal structured or decorated cospan double category is a categorified version of a commutative

Frobenius monoid. This suggests a route to categorifying the old definition of hypergraph category, Definition 7.1.

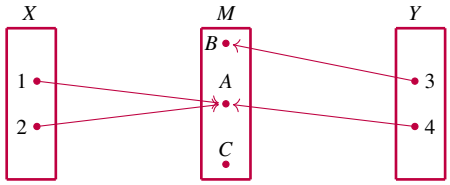
7.1. **The exoskeleton.** One common feature of decorated and structured cospans is a certain relationship between 'interfaces' and 'open systems'. Namely, in both formalisms, any cospan of interfaces can be seen as a degenerate case of an open system. For example, a typical open Petri net might look like this:



But sometimes an open Petri net has no transitions:



A Petri net with no transitions amounts to the same thing as a cospan of finite sets:



In this example finite sets are playing the role of 'interfaces', while open Petri nets are our 'open systems'. So, we are seeing that a cospan of interfaces is a degenerate open system. The only significant use of this degenerate open system is to glue other open systems together, by composition. But as we shall see, this is quite important.

We can formalize this fact as follows. First, since structured and decorated cospan categories behave so similarly, let us lump them together in the following ungainly definition:

Definition 7.2. *We define* $\mathbb{D}$ *to be a* **double category of structured/decorated cospans** *if it is either*

(1) the double category ${}_L\mathbb{C}\mathbf{sp}$ *of* structured *cospans arising from a functor* $L\colon \mathsf{A} \to \mathsf{X}$ *meeting the hypotheses of Theorem 3.1, or*

(2) the double category $F\mathbb{C}\mathbf{sp}$ *of* decorated *cospans arising from a lax monoidal pseudofunctor* $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ *meeting the hypotheses of Theorem 3.3.*

In either of these cases, we say $\mathbb{D}$ *is a* **symmetric monoidal double category of structured/decorated cospans** *if*

(1) the hypotheses of Theorem 3.2 hold, so $\mathbb{D} = {}_L\mathbb{C}\mathbf{sp}$ *is symmetric monoidal, or*

(2) the hypotheses of Theorem 3.4 hold, so $\mathbb{D} = F\mathbb{C}\mathbf{sp}$ *is symmetric monoidal.*

Second, let us note a fact about cospan double categories:

Lemma 7.3. *For any category* A *with pushouts there is a double category* $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *in which:*

- *objects are objects of* A*,*
- *tight morphisms are morphisms in* A*, composed in the same way,*
- *loose morphisms are cospans in* A*, composed via pushouts,*
- *2-cells are maps of spans in* A*, composed via pushouts.*

When A *has finite colimits,* $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *naturally has the structure of a cocartesian double category, and thus becomes symmetric monoidal using finite coproducts in* A*.*

Proof. This is a special case of Theorems 3.1 and 3.2, taking $L\colon \mathsf{A} \to \mathsf{X}$ to be the identity functor, but most of these facts were known earlier [72, 82]. □

Third, for any double category $\mathbb{D}$, define its **tight category** $\mathbb{D}_0$ to be its category of objects and tight morphisms. When $\mathbb{D}$ is a structured/decorated double category, cospans in $\mathbb{D}_0$ are what we are calling cospans of interfaces.

With these preliminaries out of the way, we can formalize the idea that a cospan of interfaces is a degenerate sort of open system:

Theorem 7.4. *Let* $\mathbb{D}$ *be a double category of structured/decorated cospans. Then there is a double functor*

$$\iota\colon \mathbb{C}\mathbf{sp}(\mathbb{D}_0) \to \mathbb{D},$$

unique up to isomorphism, that restricts to the identity functor on $\mathbb{D}_0$*. If* $\mathbb{D}$ *is symmetric monoidal, then* ι *can be given the structure of a symmetric monoidal double functor.*

Proof. We divide the proof into the three lemmas, which occupy the rest of this section. Lemma 7.5 proves existence for structured cospans, Lemma 7.6 proves existence for decorated cospans, and Lemma 7.7 proves uniqueness up to isomorphism in both cases. □

Lemma 7.5. *Let* $L\colon \mathsf{A} \to \mathsf{X}$ *be a functor,* X *a category with pushouts, and* ${}_L\mathbb{C}\mathbf{sp}$ *the double category of L-structured cospans. Then there is a double functor* $\iota\colon \mathbb{C}\mathbf{sp}(\mathsf{A}) \to {}_L\mathbb{C}\mathbf{sp}$ *such that*

- ι *acts as the identity on objects and tight morphisms,*
- ι *acts as L on loose morphisms and 2-cells: it maps any loose morphism*

$$a \xrightarrow{i} m \xleftarrow{o} b$$

in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *to the structured cospan*

$$L(a) \xrightarrow{L(i)} L(m) \xleftarrow{L(o)} L(b),$$

and it maps any 2-cell

$$\begin{array}{ccccc} a & \xrightarrow{i} & m & \xleftarrow{o} & b \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \alpha} & & \downarrow{\scriptstyle g} \\ a' & \xrightarrow{i'} & m' & \xleftarrow{o'} & b' \end{array}$$

in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *to the map of structured cospans*

$$\begin{array}{ccccc} L(a) & \xrightarrow{L(i)} & L(m) & \xleftarrow{L(o)} & L(b) \\ \downarrow{\scriptstyle L(f)} & & \downarrow{\scriptstyle L(\alpha)} & & \downarrow{\scriptstyle L(g)} \\ L(a') & \xrightarrow{L(i')} & L(m') & \xleftarrow{L(o')} & L(b'). \end{array}$$

If A *and* X *also have finite coproducts and* L *preserves them, then* $\iota\colon \mathbb{C}\mathbf{sp}(\mathsf{A}) \to {}_L\mathbb{C}\mathbf{sp}$ *can be given the structure of a symmetric monoidal double functor.*

Proof. This follows from Theorems 4.1 and 4.2 on maps between structured cospan categories, applied to this square in **Rex**:

$$\begin{array}{ccc} \mathsf{A} & \xrightarrow{1} & \mathsf{A} \\ {\scriptstyle 1}\downarrow & \Downarrow 1 & \downarrow{\scriptstyle F} \\ \mathsf{A} & \xrightarrow{F} & \mathsf{X}. \end{array}$$

□

In the decorated cospan case we use the fact that given a lax monoidal pseudofunctor $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$, any object $m \in \mathsf{A}$ has a blandest possible decoration, the **empty decoration** $v_m \in F(m)$. This is the object of $F(m)$ given by the composite

$$1 \xrightarrow{\phi_0} F(0) \xrightarrow{F(!)} F(m)$$

where ϕ_0 is the unitor for the monoidal structure on F and ! is the unique morphism from 0 to m. The empty decoration is functorial: in fact, to any morphism $\alpha\colon m \to m'$ in A we can assign a decoration morphism $F(\alpha)(v_m) \to v_{m'}$ which is simply the identity, since $F(\alpha)(v_m) = v_{m'}$.

Lemma 7.6. *Let* A *be a category with finite colimits,* $F\colon (\mathsf{A}, +) \to (\mathbf{Cat}, \times)$ *a lax monoidal pseudofunctor, and* $F\mathbb{C}\mathbf{sp}$ *the double category of* F*-decorated cospans. Then there is a double functor* $i\colon \mathbb{C}\mathbf{sp}(\mathsf{A}) \to F\mathbb{C}\mathbf{sp}$ *such that*

- ι *acts as the identity on objects and tight morphisms,*
- ι *equips any loose morphism with the empty decoration, and any 2-cell with the identity morphism of decorations: it maps any loose morphism*

$$a \xrightarrow{i} m \xleftarrow{o} b$$

in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *to the decorated cospan*

$$a \xrightarrow{i} m \xleftarrow{o} b \qquad\qquad v_m \in F(m)$$

and it maps any 2-cell

$$\begin{array}{ccccc} a & \xrightarrow{i} & m & \xleftarrow{o} & b \\ {\scriptstyle f}\downarrow & & {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle g} \\ a' & \xrightarrow{i'} & m' & \xleftarrow{o'} & b' \end{array}$$

in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *to the map of decorated cospans*

$$\begin{array}{ccccc} a & \xrightarrow{i} & m & \xleftarrow{o} & b \\ {\scriptstyle f}\downarrow & & {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle g} \\ a' & \xrightarrow{i'} & m' & \xleftarrow{o'} & b' \end{array} \qquad \begin{array}{l} v_m \in F(m) \\ \\ \\ v_{m'} \in F(m') \end{array}$$

with the decoration morphism $F(\alpha)(v_m) \to v_{m'}$ *given by the identity.*

If F *is a symmetric lax monoidal pseudofunctor, then* $\iota \colon \mathbb{C}\mathbf{sp}(\mathsf{A}) \to F\mathbb{C}\mathbf{sp}$ *can be given the structure of a symmetric monoidal double functor.*

Proof. This follows from Theorem 6.2. □

To prove the uniqueness up to isomorphism in Theorem 7.4, we can use a nice characterization of $\mathbb{C}\mathbf{sp}(\mathsf{A})$ due to Dawson, Paré and Pronk [38]. This says roughly that for any category A with pushouts, $\mathbb{C}\mathbf{sp}(\mathsf{A})$ is the free fibrant double category having A as its tight category.

To state this result more precisely, recall that a **lax** double functor between double categories, say $F \colon \mathbb{C} \to \mathbb{D}$, is like an ordinary (i.e., pseudo) double functor except that the laxator $F(f) \circ F(g) \Rightarrow F(f \circ g)$ and unitor $1 \Rightarrow F(1)$ for composition of loose morphisms are not required to be invertible. A lax double functor is **normal** if its unitor is invertible. Among the lax double functors, the normal ones are precisely those that preserve companions and conjoints [38, Prop. 3.8].

Lemma 7.7. *Let* A *be a category with pushouts and* $\mathbb{D}$ *a fibrant double category. Then composing with the inclusion* $\mathsf{A} \to \mathbb{C}\mathbf{sp}(\mathsf{A})$ *gives an equivalence between the category of normal lax double functors*

$$\mathbb{C}\mathbf{sp}(\mathsf{A}) \to \mathbb{D}$$

and the category of functors

$$\mathsf{A} \to \mathbb{D}_0.$$

Proof. This is [38, Thm. 3.15], dualized to apply to cospans. □

Using this lemma and setting $\mathsf{A} = \mathbb{D}_0$, we see that under the hypotheses of Theorem 7.4, the double functor $\iota \colon \mathbb{C}\mathbf{sp}(\mathbb{D}_0) \to \mathbb{D}$ is characterized uniquely up to isomorphism by the property that it restricts to the identity on $\mathbb{D}_0$. This completes the proof of Theorem 7.4.

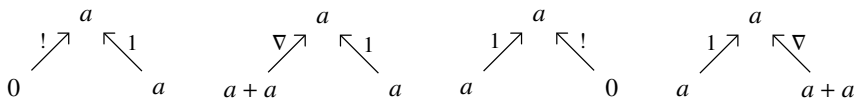
The perceptive reader will notice that most of this proof could have been quickly dispatched using Lemma 7.7. However, this lemma does not cover the symmetric monoidal aspects, and our statement of Theorem 7.4 does not claim any uniqueness for the symmetric monoidal structure of ι. It may be possible to adapt the lemma to deal with this.

Challenge 7.8. *Prove a result similar to Lemma 7.7 that applies when* A *has finite colimits, characterizing* $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *as a symmetric monoidal double category in terms of a left universal*

property. For example, perhaps $\mathbb{C}\mathbf{sp}(\mathsf{A})$ *is the free fibrant symmetric monoidal category having* A *as its tight category.*

However, we wanted to give explicit descriptions of the double functor $\iota\colon \mathbb{C}\mathbf{sp}(\mathbb{D}_0) \to \mathbb{D}$ for structured and decorated cospan categories, to make it crystal clear how cospans of interfaces can be seen as open systems in either of these formalisms.

7.2. **The outer shell.** We have shown that any symmetric monoidal structured/decorated cospan category $\mathbb{D}$ has a kind of 'exoskeleton' involving only cospans of interfaces, namely the double category $\mathbb{C}\mathbf{sp}(\mathbb{D}_0)$. However, we can trim this down further by replacing $\mathbb{D}_0$ with the free category with finite colimits on the set of objects of $\mathbb{D}_0$. This gives what could be called the 'outer shell' of $\mathbb{D}$: the double category containing only those cospans that can be defined using the objects of $\mathbb{D}$ and finite colimits. Examples include the following:



Here a is any object of $\mathbb{D}$, ! is the unique morphism from the initial object, and ∇ is the codiagonal. If we draw these four cospans as string diagrams:

we see they echo the four basic wiring patterns listed at the start of Section 7. This is no coincidence. Fong and Spivak [45] discovered that these wiring patterns, and the laws governing them, arise whenever one considers cospans in a category with finite colimits. To study such cospans in their purest form, they introduced cospans in the free category with finite colimits on an arbitrary set. However, they only studied the *category* of such cospans. Here we study the double category of such cospans.

The free category with finite colimits on one object is FinSet. More generally, for any finite set X, the free category with finite colimits on X is FinSet^X. But this fails when X is infinite. Instead, we need to use the category of finite sets equipped with a map to X, which we denote as $\mathsf{FinSet} \downarrow X$, even though X itself need not be finite.

Lemma 7.9. *The free category with finite colimits on a set* X *is* $\mathsf{FinSet} \downarrow X$*. More precisely, suppose* A *is a category with finite colimits. Then any function sending elements of* X *to objects of* A *extends to a finite-colimit-preserving functor from* $\mathsf{FinSet} \downarrow X$ *to* A*. Moreover, any two such extensions are naturally isomorphic in a unique way.*

Proof. We use a general result [56, Thm. 5.35] about the free category with colimits of a given sort on a small category C, and specialize this to the case where $\mathsf{C} = \mathsf{Disc}(X)$, the discrete category on X. This general result finds the desired free category inside the presheaf category $\widehat{\mathsf{C}} = \mathsf{Set}^{\mathsf{C}^{\mathrm{op}}}$. In particular, it says that the free category with finite colimits on C is the full subcategory of $\widehat{\mathsf{C}}$ whose objects are finite colimits of representables. In the case $\mathsf{C} = \mathsf{Disc}(\mathsf{X})$, this full subcategory is equivalent to $\mathsf{FinSet} \downarrow X$. □

Now, suppose A is a small category with finite colimits. Let

$$\tilde{\mathsf{A}} = \mathsf{FinSet} \downarrow \mathrm{Ob}(\mathsf{A})$$

where $\mathrm{Ob}(\mathsf{A})$ is the set of objects of A. By Lemma 7.9, $\tilde{\mathsf{A}}$ is the free category with colimits on the objects of A. There is a natural inclusion $\mathrm{Ob}(\mathsf{A}) \hookrightarrow \mathrm{Ob}(\tilde{\mathsf{A}})$, and thus an essentially

unique functor

$$\pi\colon \tilde{\mathsf{A}} \to \mathsf{A}$$

sending objects of A to themselves and preserving finite colimits. As one would expect, π induces a symmetric monoidal double functor

$$j\colon \mathbb{C}\mathbf{sp}(\tilde{A}) \to \mathbb{C}\mathbf{sp}(\mathsf{A}).$$

Indeed, this is a special case of Theorem 4.2.

We can summarize the story so far as follows. Let $\mathbb{D}$ be a small symmetric monoidal structured/decorated cospan category. Then there are symmetric monoidal double functors

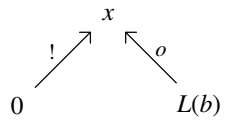
$$\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0) \xrightarrow{j} \mathbb{C}\mathbf{sp}(\mathbb{D}) \xrightarrow{\iota} \mathbb{D}.$$

We call $\mathbb{C}\mathbf{sp}(\mathbb{D})$ the 'exoskeleton' of $\mathbb{D}$, and $\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0)$ the 'outer shell'. Loose morphisms in the exoskeleton are cospans in the tight subcategory of $\mathbb{D}$, while loose morphisms in the outer shell are cospans that can be defined using only the objects of $\mathbb{D}$ and general features of categories with finite colimits. In a sense, we obtain the outer shell by stripping the exoskeleton of most of its personality, which comes from its morphisms.

We expect that this story can be carried through for a larger class of double categories, with structured and decorated cospans being just examples. Libkind and Myers' work on variable sharing theories suggests a way to generalize. We turn to this next.

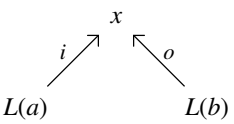
7.3. **Hypergraph double categories.** We have just seen that any symmetric monoidal structured/decorated cospan double category comes equipped with maps from two simpler double categories: its exoskeleton and its outer shell. In Section 7.4 we dig into the rich structure that still remains in the outer shell. Libkind and Myers take a complementary approach, developing a general framework to unify structured and decorated cospans, and also study open systems in the input-output paradigm [64, 71]. Here we introduce their line of thinking with the example of structured cospans.

In Section 1 we asked why an open system should have just two interfaces, and answered that this was just a matter of convenience. Libkind and Myers go further and consider open systems with just *one* interface. Suppose $\mathbb{D}$ is a symmetric monoidal structured cospan category. It turns out that we can recover all of $\mathbb{D}$ from cospans with trivial left interface:

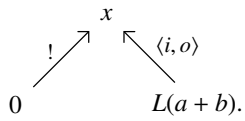


together with certain operations we can do on these: namely, tensoring them and composing them on the right with cospans coming from the outer shell.

First, we can take an arbitrary structured cospan

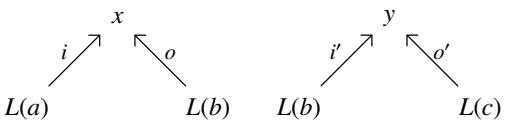


and 'fold' it to get a structured cospan with trivial left interface:



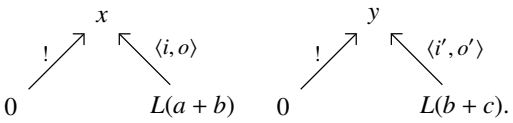
Here $\langle i, o \rangle$ is the copairing of i and o composed with the isomorphism $L(a+b) \cong L(a)+L(b)$.

Next, suppose we want to compose two structured cospans:

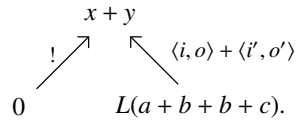


(11)

using only their folded versions:

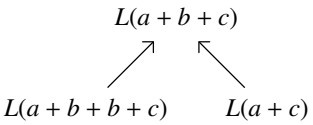


To do this we first tensor their folded versions, getting this:



(12)

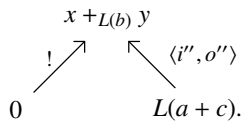
We then compose this with the following cospan:



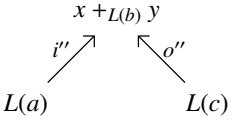
(13)

whose left leg is built using the codiagonal $\nabla \colon b+b \to b$, and whose right leg is built using the unique map $! \colon 0 \to b$.

One can show that the result of composing cospans (12) and (13) is

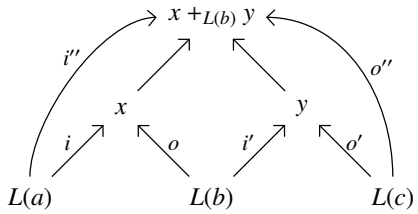


This is the folded version of the cospan

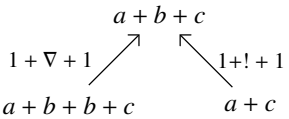


that is the composite of the cospans in Equation (11). To prove this, we do a computation showing that i'' and o'' can also be defined using the pushout diagram that defines the

desired composite:



In short, composing structured cospans in folded form requires only tensoring them and then composing the result with the cospan in Equation (13). But this cospan comes from the outer shell. To see this, note that if we take the following cospan in the outer shell $\tilde{\mathbb{D}}_0$:



and map it into $\mathbb{D}$ using the double functor discussed in Section 7.2:

$$\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0) \xrightarrow{j} \mathbb{C}\mathbf{sp}(\mathbb{D}) \xrightarrow{\iota} \mathbb{D},$$

we get the cospan in (13). The double functor j converts formal colimits to colimits in the tight category of $\mathbb{D}$; then ι applies L to everything, as in Lemma 7.5.

To generalize this sort of observation, Libkind and Myers define a 'symmetric monoidal loose right module' of a symmetric monoidal double category [64, Sec. 4]. Structured cospans with trivial left interface, and maps between these, should form a symmetric monoidal loose right module of the outer shell of $\mathbb{D}$. The argument above suggests that the whole symmetric monoidal double category $\mathbb{D}$ can be recovered from this module. The same is probably also true for decorated cospans. However, I have not seen proofs of these claims.

Challenge 7.10. *Read Libkind and Myers* [64] *and show that if $\mathbb{D}$ is a symmetric monoidal stuctured/decorated cospan double category, then loose morphisms with trivial left interface, and maps between these, give a symmetric monoidal loose right module of the outer shell $\mathbb{C}\mathbf{sp}(\tilde{\mathbb{D}}_0)$. Then show that $\mathbb{D}$ can be recovered from this symmetric monoidal loose right module.*

If the first part of this challenge can be met, every symmetric monoidal structured or decorated cospan double category wll be a 'hypergraph double category' according to the following definition, which categorifies Fong and Spivak's definition of hypergraph category [45].

Tentative Definition 7.11. *A* **hypergraph double category** *is a symmetric monoidal loose right module of any symmetric monoidal double category of the form $\mathbb{C}\mathbf{sp}(\mathsf{X})$, where X is the free category with finite colimits on some set.*

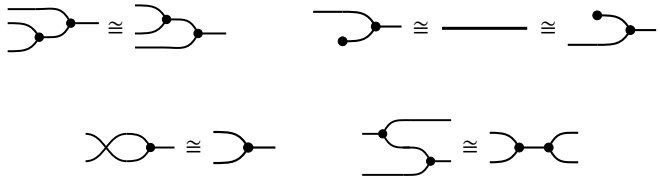
In [64, Ex. 8.11], Libkind and Myers proposed a seemingly more general definition of hypergraph double category, allowing X to be the free category with colimits on any small category. But this may be equivalent, since we can always replace X with $\tilde{\mathsf{X}}$, the free category with finite colimits on the set of objects of X, and—presumably—turn a symmetric monoidal loose right module of $\mathbb{C}\mathbf{sp}(\mathsf{X})$ into one of $\mathbb{C}\mathbf{sp}(\tilde{\mathsf{X}})$, using the symmetric monoidal double functor

$$j\colon \mathbb{C}\mathbf{sp}(\tilde{A}) \to \mathbb{C}\mathbf{sp}(\mathsf{A}).$$

7.4. **Frobenius structures.** Since any structured or decorated cospan double category comes equipped with a map from a cospan double category, some features of ordinary cospans are automatically inherited by structured and decorated cospans. For example, when $\mathbb{D}$ is a symmetric monoidal structured/decorated cospan double category, every object $a \in \mathbb{D}$ is equipped with loose morphisms

$\mu\colon a + a \nrightarrow a$ $\quad$ $\eta\colon 0 \nrightarrow a$ $\quad$ $\delta\colon a \nrightarrow a + a$ $\quad$ $\epsilon\colon a \nrightarrow 0$

that obey the laws of a special commutative Frobenius monoid up to isomorphism. Thus, the object comes with chosen invertible 2-cells



and their left-right reflected versions, and these 2-cells obey a bevy of coherence laws. All these coherence laws arise from an analysis of cospan double categories.

Before delving into this, let us begin with some generalities. First, any double category $\mathbb{D}$ has a **loose bicategory**, denoted **D**, in which:

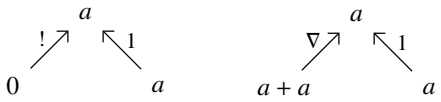
- objects are objects of $\mathbb{D}$,
- morphisms are loose morphisms of $\mathbb{D}$,
- 2-morphisms are 2-cells of $\mathbb{D}$ for which the source and target tight morphisms are identities,
- composition of morphisms is given by composition of loose morphisms in $\mathbb{D}$,
- vertical and horizontal composition of 2-morphisms are given by tight and loose composition of 2-cells in $\mathbb{D}$, respectively.

Second, any bicategory **D** has a **decategorification**, a category D in which:

- objects are objects of **D**,
- morphisms are isomorphism classes of morphisms of **D**.

Hansen and Shulman have shown that if $\mathbb{D}$ is a *fibrant* symmetric monoidal double category, **D** naturally becomes a symmetric monoidal bicategory [53, 83]. It then follows easily that D becomes a symmetric monoidal category.

Now let A be a category with finite colimits. Merely by virtue of A having finite coproducts, every object $a \in \mathsf{A}$ becomes a commutative monoid internal to $(\mathsf{A}, +)$. The unit for this monoid is the unique morphism $!\colon 0 \to a$ in A, while the multiplication is the codiagonal $\nabla\colon a \to a + a$. These morphisms give cospans in A



that we call the **unit** $\eta\colon 0 \nrightarrow a$ and **multiplication** $\mu\colon a + a \nrightarrow a$, respectively. In terms of string diagrams, these look like 'adding an extra wire that doesn't connect with anything'

and 'joining wires', respectively:

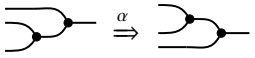
$$\eta\colon 0 \nrightarrow a \qquad\qquad \mu\colon a + a \nrightarrow a$$

Since ! and ∇ obey the commutative monoid laws, but pushouts are defined only up to isomorphism, we expect that μ and η obey the commutative monoid laws only up to coherent isomorphism. They should thus make a into a 'symmetric pseudomonoid', a concept that can be defined in any symmetric monoidal bicategory [36, Def. 17].

Because A has finite colimits, we can form the symmetric monoidal double category $\mathbb{C}\mathbf{sp}(\mathsf{A})$ as in Lemma 7.3. Since this is fibrant, Hansen and Shulman's work implies that the loose bicategory $\mathbf{Csp}(\mathsf{A})$ is symmetric monoidal. The loose morphisms μ and η live in this symmetric monoidal bicategory. And indeed, they make a into a symmetric pseudomonoid in $\mathbf{Csp}(\mathsf{A})$. In more detail:

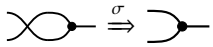
Theorem 7.12. *Let A be a category with finite colimits, and let $a \in \mathsf{A}$. The cospans $\eta\colon 0 \nrightarrow a$ and $\mu\colon a + a \nrightarrow a$ defined as above make a into symmetric pseudomonoid in $\mathbf{Csp}(\mathsf{A})$. In more detail:*

- *The multiplication μ obeys the associative law up to an isomorphism of cospans, the* **associator** $\alpha\colon \mu(\mu + 1_a) \xRightarrow{\sim} \mu(1_a + \mu)$.



  This associator can be chosen to obey the pentagon identity.
- *The unit η obeys the left and right unit laws up to isomorphisms called the* **left unitor** $\lambda\colon \mu(\eta + 1_a) \xRightarrow{\sim} 1_a$ *and* **right unitor** $\rho\colon \mu(1_a + \eta) \xRightarrow{\sim} 1_a$.

  Together with the associator, these can be chosen to obey the triangle identity.
- *The multiplication μ obeys the commutative law up to an isomorphism called the* **symmetry** $\sigma\colon \mu s \xRightarrow{\sim} \mu$, *where $s\colon a + a \nrightarrow a + a$ is the symmetry in* $\mathbf{Csp}(\mathsf{A})$.



  Together with the associator, this can be chosen to obey the hexagon identities.

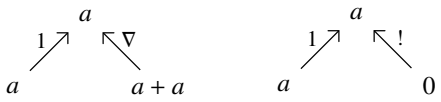
Proof Sketch. The associator, unitors and symmetry can all be defined using the universal property of pushouts, and the uniqueness clause in this universal property implies that the pentagon, triangle and hexagon diagrams commute. □

To fit this into a larger context, note that what we have done here is to turn two tight morphisms ! and ∇ in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ into their companions, η and μ. The process of turning tight morphisms into their companions can be shown to define a symmetric monoidal pseudofunctor

$$\mathrm{comp}\colon \mathsf{A} \to \mathbf{Csp}(\mathsf{A}).$$

Thus it sends commutative monoids in A to symmetric pseudomonoids in $\mathbf{Csp}(\mathsf{A})$ [36, Prop. 16].

Similarly, if we take the conjoints of $!\colon 0 \to a$ and $\nabla\colon a \to a + a$ we obtain cospans



which we call the **comultiplication** $\delta\colon a \nrightarrow a + a$ and **counit** $\epsilon\colon a \nrightarrow 0$. In terms of string diagrams, these look like 'splitting a wire' and 'capping off a wire':

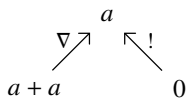
$\mu\colon a + a \nrightarrow a$ $\qquad$ $\eta\colon 0 \nrightarrow a$

Since taking conjoints is a symmetric monoidal pseudofunctor

$$\mathrm{conj}\colon \mathsf{A}^{\mathrm{op}} \to \mathbf{Csp}(\mathsf{A})$$

it sends a, which is a commutative comonoid in $(\mathsf{A}^{\mathrm{op}}, +)$, to a symmetric pseudocomonoid in $\mathbf{Csp}(\mathsf{A})$, with δ as comultiplication and ϵ as counit.

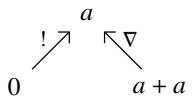
How, and why, do the pseudomonoid and pseudocomonoid structures on any object $a \in \mathbf{Csp}(\mathsf{A})$ interact? They do so because the multiplication $\mu\colon a + a \nrightarrow a$ followed by the counit $\epsilon\colon a \nrightarrow 0$ gives an important cospan called the **cup**:



which we can draw as this string diagram:

$\cup\colon a + a \nrightarrow 0.$

Similarly, we can define the **cap** to be the unit $\iota\colon 0 \nrightarrow a$ followed by the comultiplication $\delta\colon a \nrightarrow a + a$:



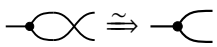
which we can draw as follows:

$\cap\colon 0 \nrightarrow a + a$

In terms of electrical circuits, these give the ability to bend wires. Mathematically they make a into its own dual in $\mathbb{C}\mathbf{sp}(\mathsf{A})$.

For any object $a \in \mathbb{C}\mathbf{sp}(\mathsf{A})$, the cap and cup obey the usual zigzag identities up to isomorphisms that in turn obey coherence laws called the swallowtail equations [85, Cor. 5.8]. But in fact, as long as the zigzag identities hold up to isomorphism, the isomorphisms can always be tweaked to make them obey the swallowtail equations [78, Thm. 2.7]. Thus, in general, we can define a **Frobenius** pseudomonoid in a monoidal bicategory to be a pseudomonoid for which the cap and cup obey the zigzag identities up to isomorphism. For an equivalent alternative definition, see [87].

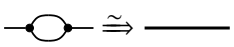
A **symmetric** Frobenius pseudomonoid is one whose underlying pseudomonoid is symmetric. One can show this implies that its underlying pseudocomonoid is also symmetric in a dual sense, involving an isomorphism



$$s\delta \cong \delta.$$

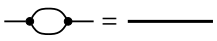
So far, we have seen that whenever A is a category with finite colimits, every object in $\mathbb{C}\mathbf{sp}(\mathsf{A})$ is a symmetric Frobenius pseudomonoid in $\mathbf{Csp}(\mathsf{A})$.

But there is more! The multiplication in this Frobenius pseudomonoid is left biadjoint to the comultiplication, with the counit ε of the biadjunction being an isomorphism:



$$\varepsilon\colon \mu\delta \cong 1_a.$$

We call a Frobenius pseudomonoid with this property **special**, extending the existing terminology where a Frobenius monoid is called special if



We do not know if there are additional coherence laws that the counit ε obeys, which should be incorporated in the definition of 'special'.

Following tradition we have spoken about pseudomonoids in a monoidal bicategory $\mathbf{D}$, which happens to be the loose bicategory of a monoidal double category $\mathbb{D}$. However, in this situation we might as well call these pseudomonoids in $\mathbb{D}$. Thus, we have shown:

Theorem 7.13. *Let A be a category with finite colimits. Then the above structures make any object $a \in \mathsf{A}$ into a special symmetric Frobenius pseudomonoid in $\mathbb{C}\mathbf{sp}(\mathsf{A})$.*

What we have seen for cospans, now follows for structured or decorated cospans:

Theorem 7.14. *Let $\mathbb{D}$ be a symmetric monoidal structured/decorated cospan double category. Then any object $a \in \mathbb{D}$ becomes a special symmetric Frobenius pseudomonoid in $\mathbb{D}$.*

Proof. Given a category A with finite colimits, we have seen how any object $a \in \mathsf{A}$ gives a special symmetric Frobenius pseudomonoid $a \in \mathbb{C}\mathbf{sp}(\mathsf{A})$. This holds in particular when $\mathsf{A} = \mathbb{D}_0$ where $\mathbb{D}$ is a symmetric monoidal structured/decorated cospan double category. Applying the symmetric monoidal pseudofunctor $\iota\colon \mathsf{Csp}(\mathbb{D}_0) \to \mathbb{D}$, we obtain a special symmetric Frobenius pseudomonoid structure on the object a regarded as an object of $\mathbb{D}$. □

At the end of Section 7.3, following ideas of Libkind and Myers [64], we proposed a definition of 'hypergraph double category' aimed at isolating the most important common features of symmetric monoidal structured and decorated cospan double categories. This definition categorified Fong and Spivak's slick definition of hypergraph category [45].

We are now in a position to pursue a different definition of hypergraph double category, which categorifies the 'old-fashioned' definition of a hypergraph category, Definition 7.1. Naively, we could define a **hypergraph double category** to be a symmetric monoidal double category $\mathbb{D}$ where each object is equipped with the structure of a special symmetric

Frobenius pseudomonoid, with these structures compatible with the monoidal structure of $\mathbb{D}$ in the sense that they obey all the equations in Definition 7.1. However, it may be overly strict to demand equations here. Furthermore, besides having a multiplication, unit, comultiplication and counit, a symmetric Frobenius pseudomonoid is equipped with extra structure, namely various 2-isomorphisms. Presumably these, too, must be compatible with the monoidal structure of $\mathbb{D}$.

Thus, there are some coherence issues to sort out, of the kind only higher category theorists truly enjoy. We leave this as a challenge:

Challenge 7.15. *Find the correct definition of 'hypergraph double category' along the following lines: it is a symmetric monoidal double category $\mathbb{D}$ where each object has the structure of a special symmetric Frobenius pseudomonoid and these structures are compatible with the symmetric monoidal structure of $\mathbb{D}$. To test the correctness of a candidate definition, show that every symmetric monoidal structured/decorated cospan category is a hypergraph double category according to this definition. If possible, also show that this definition is equivalent to the one proposed at the end of Section 7.3.*

To conclude, let us recall why this issue is important. We have some intuition of what open systems are in the variable sharing paradigm. Structured and decorated cospans gives us *examples*. But what is the right *general concept* of a symmetric monoidal double category of open systems in the variable sharing paradigm? It may be a hypergraph double category. But what, exactly, is that? We have outlined two approaches to defining this concept. Ideally they will lead to equivalent definitions.

Acknowledgements. The work described here is due to many people. I would especially like to acknowledge my collaborators in work on decorated and structured cospans and the mathematics of open systems, including Brandon Coya, Kenny Courser, Jarson Erbele, Brendan Fong, Fabrizio Genovese, Xiaoyan Li, Jade Master, Joe Moeller, Nathaniel Osgood, Sophie Libkind, Evan Patterson, Blake Pollard, Mike Shulman, Mike Stay, and Christina Vasikakopoulou. I thank Nathanael Arkor, Kevin Carlson and others on the Category Theory Community Server for helpful conversations about double categories, and David Jaz Myers for persistently pushing forward the theory of open systems.

References

[1] R. Aduddell, J. Fairbanks, A. Kumar, P. S. Ocal, E. Patterson and B. T. Shapiro, A compositional account of motifs, mechanisms, and dynamics in biochemical regulatory networks, *Compositionality* **6** (2024). Also available as arXiv:2301.01445. (Referred to on page 2, 30, 31.)

[2] E. Aleiferi, *Cartesian Double Categories with an Emphasis on Characterizing Spans*, Ph.D. Thesis, Department of Mathematics, Dalhousie University. Available as arXiv:1809.06940. (Referred to on page 8.)

[3] AlgebraicDynamics.jl, available at https://algebraicjulia.github.io/AlgebraicDynamics.jl/. (Referred to on page 36.)

[4] AlgebraicPetri.jl, available at https://algebraicjulia.github.io/AlgebraicPetri.jl/. (Referred to on page 20, 36.)

[5] A. Baas, J. Fairbanks, M. Halter, S. Libkind and E. Patterson, An algebraic framework for structured epidemic modeling, *Phil. Trans. Roy. Soc. A* **380** (2022), 20210309. Also available as arXiv:2203.16345. (Referred to on page 20, 31.)

[6] J. C. Baez and A. Chaudhuri, Graphs with polarities. Available as arXiv:2506.23375. (Referred to on page 2, 9.)

[7] J. C. Baez and K. Courser, Structured cospans, *Theory Appl. Categ.* **35** (2020), 1771–1822. Also available as arXiv:1911.04630. (Referred to on page 2, 8, 9, 18.)

[8] J. C. Baez and K. Courser, Coarse-graining open Markov processes, *Theory Appl. Categ.* **33** (2018), 1223–1268. Also available as arXiv:1710.11343. (Referred to on page 2.)

[9] J. C. Baez, K. Courser and C. Vasilakopoulou, Structured versus decorated cospans, *Compositionality* **4** 3 (2022). Also available as arXiv:2101.09363. (Referred to on page 9, 10, 11, 12, 30, 35.)

[10] J. C. Baez, B. Coya and F. Rebro, Props in circuit theory, *Theory Appl. Categ.* **33** (2018), 727–783. Available as arXiv:1707.08321. (Referred to on page 2.)

[11] J. C. Baez and J. Erbele, Categories in control, *Theory Appl. Categ.* **30** (2015), 836–881. Also available as arXiv:1405.6881. (Referred to on page 2, 27.)

[12] J. C. Baez and B. Fong, A compositional framework for passive linear networks, *Theory Appl. Categ.* **33** (2018), 1158–1222. Also available as arXiv:1504.05625. (Referred to on page 2, 38.)

[13] J. C. Baez, B. Fong and B. S. Pollard, A compositional framework for Markov processes, *Jour. Math. Phys.* **57** (2016), 033301. Also available as arXiv:1508.06448. (Referred to on page 2.)

[14] J. C. Baez, F. Genovese, J. Master and M. Shulman, Categories of nets, in *36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, Rome, Italy, 2021, pp. 1–13. Also available as arXiv:2101.04238. (Referred to on page 13, 20, 21.)

[15] J. C. Baez and A. Lauda, A prehistory of n-categorical physics, in *Deep Beauty: Mathematical Innovation and the Search for an Underlying Intelligibility of the Quantum World*, ed. Hans Halvorson, Cambridge U. Press, Cambridge, 2011, pp. 13-128. Also available as arXiv:0908.2469. (Referred to on page 2, 3.)

[16] J. C. Baez, X. Li, S. Libkind, N. D. Osgood and E. Redekopp, A categorical framework for modeling with stock and flow diagrams, in *Mathematics of Public Health: Mathematical Modelling from the Next Generation*, eds. J. David and J. Wu, Springer, 2003, pp. 175–207. Also available as arXiv:2211.01290. (Referred to on page 2, 36.)

[17] J. C. Baez, X. Li, S. Libkind, N. D. Osgood and E. Patterson, Compositional modeling with stock and flow diagrams, *Electron. Proc. Theor. Comput. Sci.* **380** (2023), 77–96. Also available as arXiv:2205.08373. (Referred to on page 2, 36.)

[18] J. C. Baez, O. Lynch and J. Moeller, Compositional thermostatics, *J. Math. Phys.* **64** (2023), 023304. Also available as arXiv:2111.10315. (Referred to on page 2.)

[19] J. C. Baez and J. Master, Open Petri nets, *Math. Struct. Comput. Sci.* **30** (2020), 314–341. Also available as arXiv:1808.05415. (Referred to on page 13, 16, 17.)

[20] J. C. Baez and B. S. Pollard, A compositional framework for chemical reaction networks, *Rev. Math. Phys.* **29** (2017), 1750028. Also available as arXiv:1704.02051. (Referred to on page 24, 29, 35.)

[21] J. C. Baez and M. Stay, Physics, topology, logic and computation: a Rosetta Stone, in *New Structures for Physics*, ed. B. Coecke, Lecture Notes in Physics 813, Springer, Berlin, 2011, pp. 95–172. Also available as arXiv:0903.0340. (Referred to on page 2, 5.)

[22] J. C. Baez, D. Weisbart and A. M. Yassine, Open systems in classical mechanics, *Jour. Math. Phys.* **62** (2021), 042902. Also available as arXiv:1710.11392. (Referred to on page 2, 27.)

[23] F. Bonchi, P. Sobociński and F. Zanasi, A categorical semantics of signal flow graphs, in *CONCUR 2014–Concurrency Theory*, eds. P. Baldan and D. Gorla, Lecture Notes in Computer Science **8704**, Springer, Berlin, 2014, pp. 435–450. Also available at http://users.ecs.soton.ac.uk/ps/papers/sfg.pdf. (Referred to on page 2.)

[24] A. Carboni, Matrices, relations, and group representations, *J. Algebra* **136** (1991), 497–529. (Referred to on page 39.)

[25] CatColab. Available at https://catcolab.org/help. (Referred to on page 37.)

[26] Catlab.jl. Available at https://github.com/AlgebraicJulia/Catlab.jl. (Referred to on page 11, 36.)

[27] K. Courser, A bicategory of decorated cospans, *Theory Appl. Categ.* **32** (2017), 995–1027. Also available as arXiv:1605.08100. (Referred to on page 6.)

[28] K. Courser, *Open Systems: a Double Categorical Perspective*, Ph.D. thesis, Department of Mathematics, U. C. Riverside, 2020. Available as arXiv:2008.02394. (Referred to on page 7, 8, 9.)

[29] B. Coya, B. Fong, Corelations are the prop for extraspecial commutative Frobenius monoids, *Theory Appl. Categ.* **32** (2017), 380–395. Also available as arXiv:1601.02307. (Referred to on page .)

[30] B. Coya, A compositional framework for bond graphs. Available as arXiv:1710.00098. (Referred to on page 2, 27.)

[31] B. Coya, *Circuits, Bond Graphs, and Signal-Flow Diagrams: A Categorical Perspective*, Ph.D. thesis, Department of Mathematics, U. C. Riverside, 2018. Available as arXiv:1805.08290. (Referred to on page 2, 27.)

[32] G. Craciun, Toric differential inclusions and a proof of the Global Attractor Conjecture. Available as arXiv:1501.02860. (Referred to on page 37.)

[33] G. Craciun, M. Mincheva, C. Pantea and P. Y. Yu, A graph-theoretic condition for delay stability of reaction systems. Available as arXiv:2105.07321. (Referred to on page 37.)

[34] G. Craciun, F. Nazarov and C. Pantea, Persistence and permanence of mass-action and power-law dynamical systems, *SIAM J. Appl. Math.* **73** (2013), 305–329. (Referred to on page 37.)
[35] G. Craciun, Y. Tang and M. Feinberg, Understanding bistability in complex enzyme-driven reaction networks, *PNAS* **103** (2006), 8697–8702. (Referred to on page 31, 37.)
[36] B. Day and R. Street, Monoidal bicategories and Hopf algebroids, *Adv. Math.* **129** (1997), 99–157. (Referred to on page 49.)
[37] R. J. M. Dawson, B. Paré and D. A. Pronk, Universal properties of Span, *Theory Appl. Categ.* **13** (2004), 61–85. Available at http://www.tac.mta.ca/tac/volumes/13/4/13-04abs.html. (Referred to on page .)
[38] R. J. M. Dawson, B. Paré and D. A. Pronk, The span construction, *Theory Appl. Categ.* **24** (2010), 302–377. Available at http://www.tac.mta.ca/tac/volumes/24/13/24-13abs.html. (Referred to on page 2, 43.)
[39] M. Feinberg, Chemical reaction network structure and the stability of complex isothermal reactors: I. The deficiency zero and deficiency one theorems, *Chem. Eng. Sci.* **42** (1987), 2229–2268. (Referred to on page 37.)
[40] M. Feinberg, *Foundations of Chemical Reaction Network Theory*, Springer, Berlin, 2019. (Referred to on page 37.)
[41] J. L. Fiadeiro and V. Schmitt, Structured co-spans: an algebra of interaction protocols, in *International Conference on Algebra and Coalgebra in Computer Science*, eds. T. Mossakowski, U. Montanari and M. Haveraaen, Lecture Notes in Computer Science 4624, Springer, Berlin, 2007, pp. 194–208. (Referred to on page 9.)
[42] B. Fong, Decorated cospans, *Theory Appl. Categ.* **30** (2015), 1096–1120. Also available as arXiv:1502.00872. (Referred to on page 9, 11, 38.)
[43] B. Fong, *The Algebra of Open and Interconnected Systems*, Ph.D. thesis, Department of Computer Science, University of Oxford, 2016. Also available as arXiv:1609.05382. (Referred to on page 7, 11, 38.)
[44] B. Fong, P. Rapisarda and P. Sobocinski, A categorical approach to open and interconnected dynamical systems, in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, New York, 2016, pp. 1–10. Available as arXiv:1510.05076. (Referred to on page 2.)
[45] B. Fong and D. Spivak, Hypergraph categories, *J. Pure Appl. Alg.* **223** (2019), 4746–4777. Also available as https://arxiv.org/abs/1806.08304. (Referred to on page 39, 44, 47, 51.)
[46] F. Gadducci and R. Heckel, An inductive view of graph transformation, in *Recent Trends in Algebraic Development Techniques (Tarquinia, 1997)*, Springer Lecture Notes in Computer Science, vol. 1376, Springer, Berlin, 1998, pp. 223–237. (Referred to on page 39.)
[47] C. Girault and R. Valk, *Petri Nets for Systems Engineering: a Guide to Modeling, Verification, and Applications*, Springer, Berlin, 2013. (Referred to on page 13.)
[48] M. Grandis and R. Paré, Limits in double categories, *Cah. Top. Géom. Diff.* **40** (1999), 162–220. Available at https://www.numdam.org/item/CTGDC_1999_40_3_162_0/ (Referred to on page 2.)
[49] M. Grandis and R. Paré, Adjoints for double categories, *Cah. Top. Géom. Diff.* **45** (2004), 193–240. (Referred to on page 2.)
[50] R. Gordon, A. J. Power and R. Street, Coherence for tricategories, *Mem. Amer. Math. Soc.* **558**, 1995. (Referred to on page 6.)
[51] P. J. Haas, *Stochastic Petri Nets: Modelling, Stability, Simulation*, Springer, Berlin, 2002. (Referred to on page 31.)
[52] M. Halter and E. Patterson, Compositional epidemiological modeling using structured cospans, 2020. Available at https://www.algebraicjulia.org/blog/post/2020/10/structured-cospans. (Referred to on page 11.)
[53] L. W. Hansen and M. Shulman, Constructing symmetric monoidal bicategories functorially. Available as arXiv:1910.09240. (Referred to on page 48.)
[54] C. Heunen and J. Vicary, *Categories for Quantum Theory: an Introduction*, Oxford U. Press, Oxford, 2019. (Referred to on page 5.)
[55] F. Horn and R. Jackson, General mass action kinetics, *Arch. Ration. Mech. Anal.* **47** (1972), 81–116. (Referred to on page 37.)
[56] G. M. Kelly, *Basic Concepts of Enriched Category Theory*, Cambridge U. Press, Cambridge, 1982. Also available at http://www.tac.mta.ca/tac/reprints/articles/10/tr10abs.html. (Referred to on page 44.)
[57] A. Kissinger, Finite matrices are complete for (dagger-)hypergraph categories. Available as arXiv:1406.5942. (Referred to on page 39.)
[58] I. Koch, Petri nets—a mathematical formalism to analyze chemical reaction networks, *Mol. Inform.* **29** (2010), 838–843. (Referred to on page 31.)
[59] J. Kock, *Frobenius Algebras and 2D Topological Quantum Field Theories*, Cambridge U. Press, Cambridge, 2003. Short version available at https://mat.uab.cat/k̃ock/TQFT/FS.pdf. (Referred to on page 3.)

[60] J. Kock, Whole-grain Petri nets and processes, *J. ACM* **70** (2022), 1–58. Also available as arXiv:2005.05108. (Referred to on page 2, 20.)
[61] F. W. Lawvere, *Functorial Semantics of Algebraic Theories*, Ph.D. thesis, Columbia University, 1963. Reprinted in *Theory Appl. Categ.* **5** (2004), 1–121. Available at . (Referred to on page 2.)
[62] X. Li, P. L. Mabry, N. D. Osgood and E. Patterson, Compositional system dynamics: the higher mathematics underlying system dynamics diagrams & practice. Available at arXiv:2509.18475. (Referred to on page 36.)
[63] S. Libkind, An algebra of resource sharing machines. Available as arxiv:2007.14442. (Referred to on page 2, 37.)
[64] S. Libkind and D. J. Myers, Towards a double operadic theory of systems. Available as arXiv:2505.18329. (Referred to on page 2, 37, 39, 45, 47, 51.)
[65] O. Lynch, *Relational Composition of Physical Systems: A Categorical Approach*, M.Sc. Thesis, Department of Physics, Universiteit Utrecht, 2022. Available at arXiv:2310.06088. (Referred to on page 2, 27.)
[66] J. Master, Petri nets based on Lawvere theories, *Math. Struct. Comp. Sci.* **30** (2020), 833–864. Also available as arXiv:1904.09091. (Referred to on page 15.)
[67] P. McCrudden, Balanced coalgebroids, *Theory Appl. Categ.* **7** (2000), 71–147. Available at http://www.tac.mta.ca/tac/volumes/7/n6/7-06abs.html. (Referred to on page 6.)
[68] ModelCollab. Available at https://modelcollab.usask.ca/. (Referred to on page 36.)
[69] J. Morton, Belief propagation in monoidal categories, *Electron. Proc. Theor. Comput. Sci.* **172** (2014) 262–269. Also available as arXiv:1504.2618. (Referred to on page 39.)
[70] D. J. Myers, Double categories of open dynamical systems (extended abstract), *EPTCS* **333** (2021), 154–167. Available as arXiv:2005.05956. (Referred to on page 2.)
[71] D. J. Myers, *Categorical Systems Theory*, draft as of September 3, 2023. Available at https://www.davidjaz.com/Papers/DynamicalBook.pdf. (Referred to on page 2, 37, 45.)
[72] S. Niefield, Span, cospan, and other double categories, *Theory Appl. Categ.* **26** (2012), 729–742. Available as arXiv:1201.3789. (Referred to on page 41.)
[73] B. Paré, Superspans, talk at the Octoberfest, Ottawa, 2015. Available at https://www.mscs.dal.ca/ pare/Superspans(Beamer).pdf. (Referred to on page 9.)
[74] E. Patterson, Structured and decorated cospans from the viewpoint of double category theory, *Electron. Proc. Theor. Comput. Sci.* **397** (2023). Also available as arXiv:2304.00447. (Referred to on page 8, 9.)
[75] E. Patterson and M. Halter, Compositional epidemiological modeling using structured cospans, *AlgebraicJulia Blog*, 2020. Available at https://blog.algebraicjulia.org/post/2020/10/structured-cospans/. (Referred to on page 36.)
[76] J. L. Peterson, *Petri Net Theory and the Modeling of Systems*, Prentice-Hall, New Jersey, 1981. (Referred to on page 13.)
[77] B. S. Pollard, *Open Markov Processes and Reaction Networks*, Ph.D. thesis, Department of Physics, U. C. Riverside, 2017. Available as arXiv:1709.09743. (Referred to on page 7.)
[78] P. Pstrągowski, On dualizable objects in monoidal bicategories. Available as arXiv:1411.6691. (Referred to on page 50.)
[79] R. Rosebrugh, N. Sabadini, R. F. C. Walters, Generic commutative separable algebras and cospans of graphs, *Theory Appl. Categ.* **15** (2005), 164–177. Available at http://tac.mta.ca/tac/volumes/15/6/15-06abs.html. (Referred to on page 39.)
[80] P. Selinger, Autonomous categories in which $A \cong A^*$ (extended abstract), talk at QPL 2010. Available at https://ncatlab.org/nlab/files/SelingerSelfDual.pdf. (Referred to on page 5.)
[81] P. Selinger, A survey of graphical languages for monoidal categories, in *New Structures for Physics*, ed. B. Coecke, Lecture Notes in Physics **813**, Springer, Berlin, 2011. Also available as arXiv:0908.3347. (Referred to on page 5.)
[82] M. Shulman, Framed bicategories and monoidal fibrations, *Theory Appl. Categ.* **20** (2008), 650–738. Available at http://www.tac.mta.ca/tac/volumes/20/18/20-18abs.html. Also available as arXiv:0706.1286. (Referred to on page 2, 41.)
[83] M. Shulman, Constructing symmetric monoidal bicategories. Available as arXiv:1004.0993. (Referred to on page 2, 6, 48.)
[84] D. Spivak, The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits, 2017. Available as arxiv:1512.01602. (Referred to on page 5.)
[85] M. Stay, Compact closed bicategories, *Theory Appl. Categ.* **31** (2016), 755–798. Available at http://www.tac.mta.ca/tac/volumes/31/26/31-26abs.html. Also available as arXiv:1301.1053. (Referred to on page 6, 50.)
[86] StockFlow.jl. Available at https://algebraicjulia.github.io/StockFlow.jl/. (Referred to on page 36.)

[87] R. Street, Frobenius monads and pseudomonoids, *J. Math. Phys.* **45** (2004), 3930–3948. (Referred to on page 50.)
[88] D. J. Wilkinson, *Stochastic Modelling for Systems Biology*, Taylor and Francis, New York, 2006. (Referred to on page 31.)
[89] R. J. Wood, Abstract proarrows I, *Cahiers* **23** 3 (1982), 279–290. Available at https://www.numdam.org/item/CTGDC_1982__23_3_279_0/ . (Referred to on page 2.)
[90] R. J. Wood, Proarrows II, *Cahiers* **26** 2 (1985) 135–168. Available at https://www.numdam.org/item/CTGDC_1985__26_2_135_0/ . (Referred to on page 2.)
[91] D. Yau, *Operads of Wiring Diagrams*, Lecture Notes in Mathematics **2192**, Springer, Berlin, 2018. Also available at arXiv:1512.01602. (Referred to on page 5.)