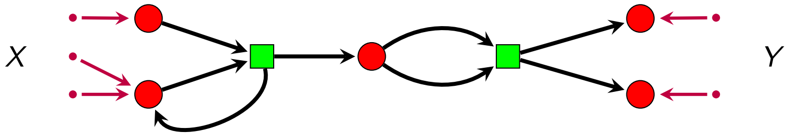
The Mathematics of Open Reaction Networks

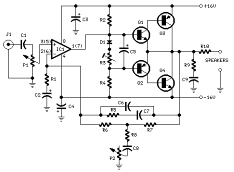


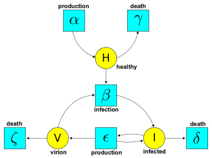
John Baez
U. C. Riverside / Centre for Quantum Technologies

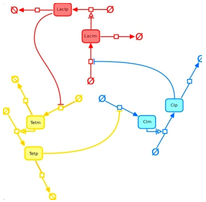
Dynamics, Thermodynamics and Information Processing in Chemical Networks

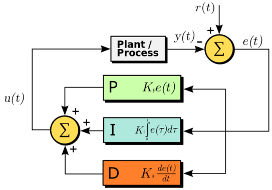
June 13, 2017

In many areas of science and engineering, people use *networks*, drawn as boxes connected by wires:





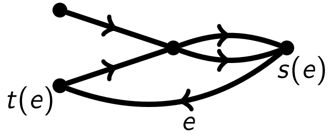




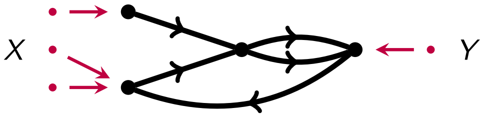
We need a good general theory of these!

Networks come in many kinds, but they share some general features.

Let's illustrate them with the simplest case, where a network is just a **graph**: a set V of vertices, a set E of edges, and maps $s, t\colon E \to V$ assigning each edge its **source** and **target**:



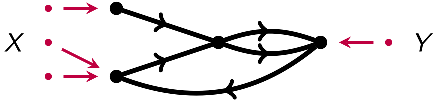
Networks with specified inputs and outputs let us describe *open systems*, meaning systems where stuff can flow in or out:



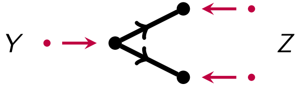
Beware: we don't require that stuff flow *in* at the 'inputs' and *out* at the 'outputs'.

The distinction between inputs and outputs is still useful. We can stick together two open systems by attaching the outputs of the first to the inputs of the second, if they match.

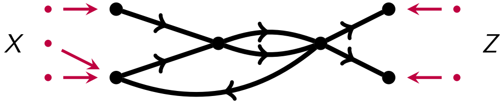
We write a network with inputs X and outputs Y as $f\colon X \to Y$.
We can **compose** $f\colon X \to Y$



and $g\colon Y \to Z$



obtaining $gf\colon X \to Z$, as follows:



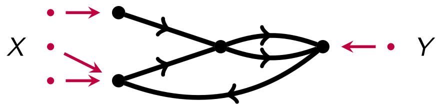
Composition obeys the associative law:

$$(hg)f = h(gf)$$

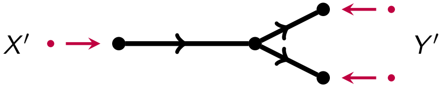
and every object has a morphism $1_X \colon X \to X$ that's the identity for composition. So, we get a **category** with networks of our chosen kind as morphisms.

But we get more, since we can also set networks side by side, or 'tensor' them.

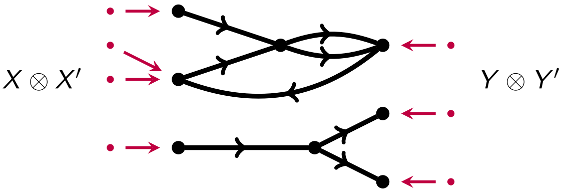
Given $f\colon X \to Y$



and $g\colon X' \to Y'$



we tensor them to get $f \otimes g\colon X \otimes X' \to Y \otimes Y'$ as follows:



Composition and tensoring obey some well-known rules, such as

$$(g \otimes g')(f \otimes f') = gf \otimes g'f'$$

These rules form the definition of a *monoidal* category.

Monoidal categories let us study the *behavior* of open systems using Lawvere's idea of 'functorial semantics'. This is already popular in computer science.

There's a category where the morphisms are programs in a given language. The map sending each program to its behavior is a functor.

The same idea works for chemical reaction networks!

Given categories **C** and **D**, a **functor** $F\colon \mathbf{C} \to \mathbf{D}$ does this:

- It sends each object X of **C** to an object $F(X)$ of **D**.
- It sends each morphism $f\colon X \to Y$ in **C** to a morphism $F(f)\colon F(X) \to F(Y)$ in **D**.
- It preserves composition: $F(gf) = F(g)F(f)$.
- It preserves identities: $F(1_X) = 1_{F(X)}$.

A **monoidal** functor between monoidal categories also preserves tensor products: $F(f \otimes g) = F(f) \otimes F(g)$.

If you pick a computer language, there's a monoidal category $\mathbf{C}$ in which:

- an object is a data type;
- a morphism $f\colon X \to Y$ is a program that takes data of type X as input and gives data of type Y as output;
- composing $f\colon X \to Y$ and $g\colon Y \to Z$ gives a program $gf\colon X \to Z$ that does first f and then g;
- tensoring $f\colon X \to Y$ and $g\colon X' \to Y$ gives a program $f \otimes g\colon X \otimes X' \to Y \otimes Y'$ that does f and g in parallel.

If each program computes some function, we get a monoidal functor $F\colon \mathbf{C} \to \mathbf{Set}$, where $\mathbf{Set}$ is the category where:

- objects are sets,
- morphisms are functions.

In reality not all programs halt, so we get $F\colon \mathbf{C} \to \mathbf{Set}_{\mathbf{part}}$, where the morphisms are *partially defined* functions.

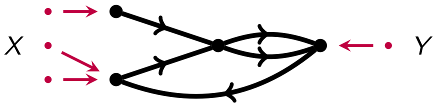
We can apply the same philosophy — functorial semantics — to networks of many kinds:

- Networks of a chosen kind, with specified input and outputs, will be morphisms in a monoidal category $\mathbf{C}$.
- The 'behavior' of these networks will be described by a monoidal functor $F \colon \mathbf{C} \to \mathbf{D}$ where $\mathbf{D}$ is a monoidal category good for semantics, e.g. $\mathbf{Set}$.
- We can also use monoidal functors to describe ways of converting one kind of network to another — thus developing a unified theory of networks.

To carry out this program we can use decorated cospans:

- Brendan Fong, *The Algebra of Open and Interconnected Systems*, Ph.D. thesis, Oxford University, 2016.

For example, this:



is really a cospan of finite sets:

$$\begin{tikzcd} & V & \\ X \arrow[ur, "i"] & & Y \arrow[ul, "o"'] \end{tikzcd}$$

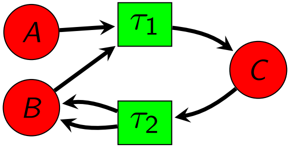
where V is 'decorated' with extra structure making it into the set of vertices of a graph: $E \overset{s}{\underset{t}{\rightrightarrows}} V$.

To make a monoidal category of open reaction networks, it's best to take any reaction network:

$$A + B \xrightarrow{\tau_1} C$$

$$C \xrightarrow{\tau_2} 2B$$

and draw it as a **Petri net**:

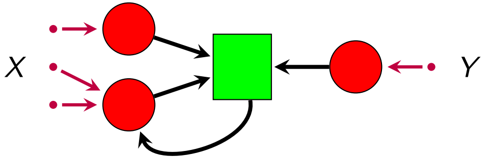


A **Petri net** is a bipartite graph. It has two kinds of vertices, **places** and **transitions**, or for chemists, **species** and **reactions**. An edge can only go from one kind of vertex to the other kind.

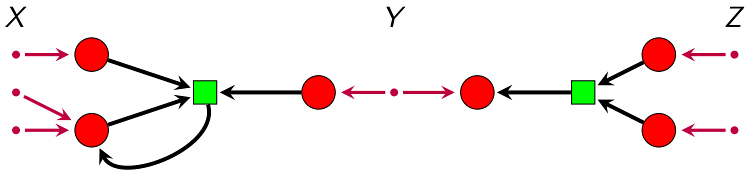
Theorem (JB–Blake Pollard)

There is a monoidal category **RNet** *where:*

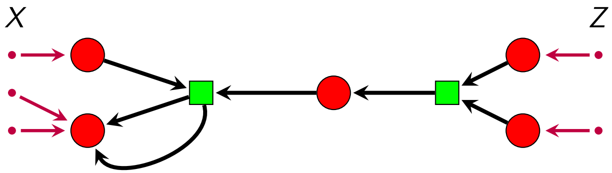
- *an object is a finite set;*
- *a morphism $f : X \to Y$ is an* **open reaction network***: a Petri net together with functions from X and Y to its set of species:*



- To compose $f\colon X \to Y$ and $g\colon Y \to Z$



we attach the outputs of f to the inputs of g:

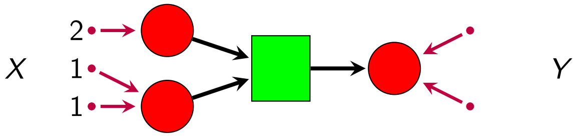


- To tensor open reaction networks, we set them side by side.

Reachability Semantics

Given an open reaction network $f\colon X \to Y$, we can ask whether $a \in \mathbb{N}^X$ can be carried to $b \in \mathbb{N}^Y$ by a series of reactions. If so, we say b is **reachable** from a.

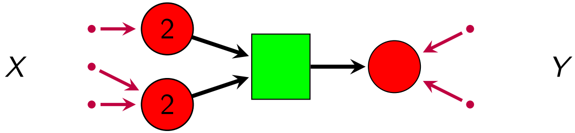
In this example $b = (0, 2)$ is reachable from $a = (2, 1, 1)$



Reachability Semantics

Given an open reaction network $f\colon X \to Y$, we can ask whether $a \in \mathbb{N}^X$ can be carried to $b \in \mathbb{N}^Y$ by a series of reactions. If so, we say b is **reachable** from a.

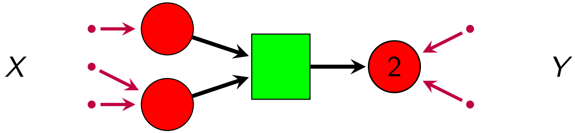
In this example $b = (0, 2)$ is reachable from $a = (2, 1, 1)$



Reachability Semantics

Given an open reaction network $f\colon X \to Y$, we can ask whether $a \in \mathbb{N}^X$ can be carried to $b \in \mathbb{N}^Y$ by a series of reactions. If so, we say b is **reachable** from a.

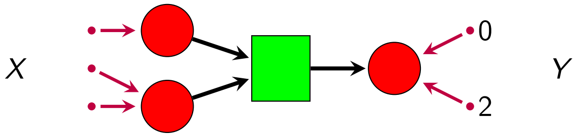
In this example $b = (0, 2)$ is reachable from $a = (2, 1, 1)$



Reachability Semantics

Given an open reaction network $f\colon X \to Y$, we can ask whether $a \in \mathbb{N}^X$ can be carried to $b \in \mathbb{N}^Y$ by a series of reactions. If so, we say b is **reachable** from a.

In this example $b = (0, 2)$ is reachable from $a = (2, 1, 1)$



Each open reaction network $f\colon X \to Y$ determines a **reachability relation**

$$F(f) = \{(a,b) : \; b \text{ is reachable from } a\} \subseteq \mathbb{N}^X \times \mathbb{N}^Y$$

Theorem (JB)

Let **Rel** *be the category where:*

- *objects are sets,*
- *a morphism from S to T is a relation $R \subseteq S \times T$.*

There is a monoidal functor

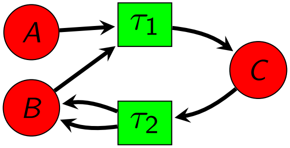
$$F\colon \mathbf{RNet} \to \mathbf{Rel}$$

sending each finite set X to $\mathbb{N}^X$ and each open reaction network $f\colon X \to Y$ to the reachability relation $F(f) \subseteq \mathbb{N}^X \times \mathbb{N}^Y$.

Rate Equation Semantics

Computer scientists like the reachability semantics. But for chemists, the 'behavior' of a reaction network with rates is often described by its *rate equation*. To write down this equation we need to choose for each reaction a positive real number called its **rate constant**.

For example, this Petri net:

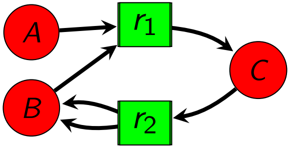


becomes a **Petri net with rates** if we choose $r_1, r_2 > 0$.

Rate Equation Semantics

Computer scientists like the reachability semantics. But for chemists, the 'behavior' of a reaction network with rates is often described by its *rate equation*. To write down this equation we need to choose for each reaction a positive real number called its **rate constant**.

For example, this Petri net:

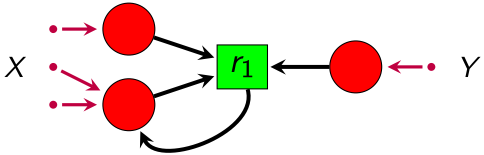


becomes a **Petri net with rates** if we choose $r_1, r_2 > 0$.

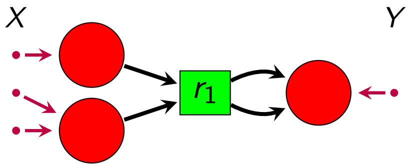
Theorem (JB–Blake Pollard)

There is a monoidal category **RxNet** *where:*

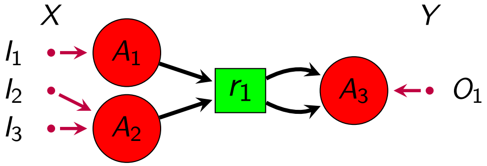
- *an object is a finite set;*
- *a morphism $f : X \to Y$ is an* **open reaction network with rates***: a Petri net with rates together with functions from X and Y to its set of species:*



An open reaction network with rates $f\colon X \to Y$ gives an **open rate equation** involving flows in and out, which can be arbitrary smooth functions of time. For example this:



An open reaction network with rates $f \colon X \to Y$ gives an **open rate equation** involving flows in and out, which can be arbitrary smooth functions of time. For example this:



gives:

$$\frac{dA_1}{dt} = -r_1\, A_1 A_2 + I_1(t)$$

$$\frac{dA_2}{dt} = -r_1\, A_1 A_2 + I_2(t) + I_3(t)$$

$$\frac{dA_3}{dt} = 2 r_1\, A_1 A_2 - O_1(t)$$

There is a monoidal category **Dynam** where:

- an object is a finite set;
- a morphism $f\colon X \to Y$ is an **open dynamical system**, meaning a cospan of finite sets

$$\begin{tikzcd} & S & \\ X \arrow[ur, "i"] & & Y \arrow[ul, "o"'] \end{tikzcd}$$

  equipped with a smooth vector field v on $\mathbb{R}^S$.

Given input and output flows $I(t) \in \mathbb{R}^X$, $O(t) \in \mathbb{R}^Y$, an open dynamical system describes how $A(t) \in \mathbb{R}^S$ changes with time:

$$\frac{d}{dt}A(t) = v(A(t)) + i_*(I(t)) - o_*(O(t))$$

where i_*, o_* push forward $\mathbb{R}$-valued functions from X, Y to S. The open rate equation is of this form.

Theorem (JB–Blake Pollard)

There is a monoidal functor $\blacksquare \colon$ **RxNet** $\to$ **Dynam** *sending any open Petri net with rates to its open dynamical system.*

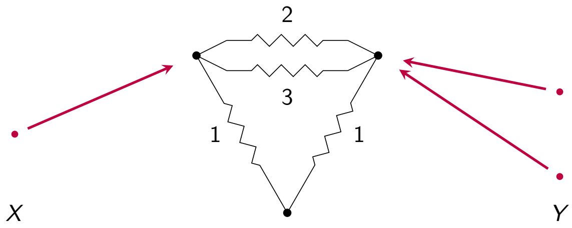
So, we can determine the rate equation of a reaction network with rates by breaking it down into simpler *open* reaction network nets with rates, and repeatedly using:

$$\blacksquare(fg) = \blacksquare(f)\,\blacksquare(g)$$

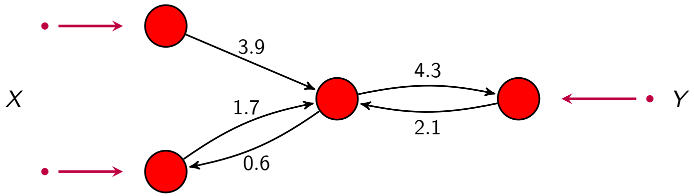
$$\blacksquare(f \otimes g) = \blacksquare(f) \otimes \blacksquare(g)$$

The same methods let us study many kinds of networks — and fit them into a unified theory of networks.

- **Electrical circuits**: JB and Brendan Fong, A compositional framework for passive linear networks.

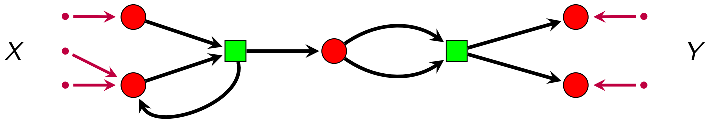


- **Markov processes**: JB, Brendan Fong and Blake Pollard, A compositional framework for Markov processes.

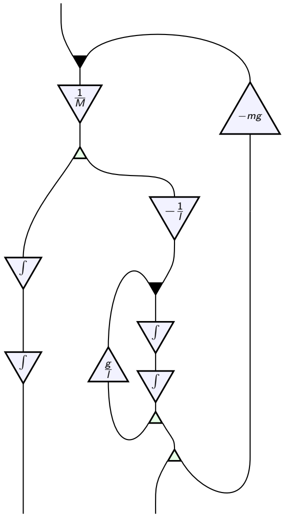


- **Reaction networks**: JB and Blake Pollard, A compositional framework for reaction networks.

  And coming soon: Blake Pollard, *Open Markov Processes and Reaction Networks*, Ph.D. Thesis, U. C. Riverside, 2017.



- **Signal-flow graphs in control theory**:
  Jason Erbele, *Categories in Control: Applied PROPs*, Ph.D. Thesis, U. C. Riverside, 2016.



We find a 'network of networks': interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}\mathbf{RxNet} \arrow[d, "\blacksquare"] \\ \mathbf{Dynam}\end{tikzcd}$$

We find a 'network of networks': interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{RxNet} \arrow[d, "{\color{gray}\blacksquare}"] \\
\mathbf{Dynam} \arrow[d, "\blacksquare"] \\
\mathbf{Rel}
\end{tikzcd}$$

We find a 'network of networks': interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{Markov} \arrow[r] & \mathbf{RxNet} \arrow[d, "\square"] \\
 & \mathbf{Dynam} \arrow[d, "\blacksquare"] \\
 & \mathbf{Rel}
\end{tikzcd}$$

We find a ‘network of networks’: interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{ResCirc} \arrow[r] & \mathbf{Markov} \arrow[r] & \mathbf{RxNet} \arrow[d, "{\color{gray}\blacksquare}"] \\
 & & \mathbf{Dynam} \arrow[d, "\blacksquare"] \\
 & & \mathbf{Rel}
\end{tikzcd}$$

We find a ‘network of networks’: interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{ResCirc} \arrow[r] \arrow[d] & \mathbf{Markov} \arrow[r] & \mathbf{RxNet} \arrow[d, "\textcolor{gray}{\blacksquare}"] \\
\mathbf{Circ} & & \mathbf{Dynam} \arrow[d, "\blacksquare"] \\
& & \mathbf{Rel}
\end{tikzcd}$$

We find a ‘network of networks’: interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{ResCirc} \arrow[r] \arrow[d] & \mathbf{Markov} \arrow[r] \arrow[ddl] & \mathbf{RxNet} \arrow[d, "{\color{gray}\blacksquare}"] \\
\mathbf{Circ} \arrow[d] & & \mathbf{Dynam} \arrow[d, "\blacksquare"] \\
\mathbf{SigFlow} & & \mathbf{Rel}
\end{tikzcd}$$

We find a 'network of networks': interesting monoidal categories connected by monoidal functors:

$$\begin{tikzcd}
\mathbf{ResCirc} \arrow[r] \arrow[d] & \mathbf{Markov} \arrow[r] \arrow[ddl] & \mathbf{RxNet} \arrow[d, "{\color{lightgray}\blacksquare}"] \\
\mathbf{Circ} \arrow[d] & & \mathbf{Dynam} \arrow[d, "\blacksquare"] \\
\mathbf{SigFlow} \arrow[rr, "\text{steady states}"'] & & \mathbf{Rel}
\end{tikzcd}$$