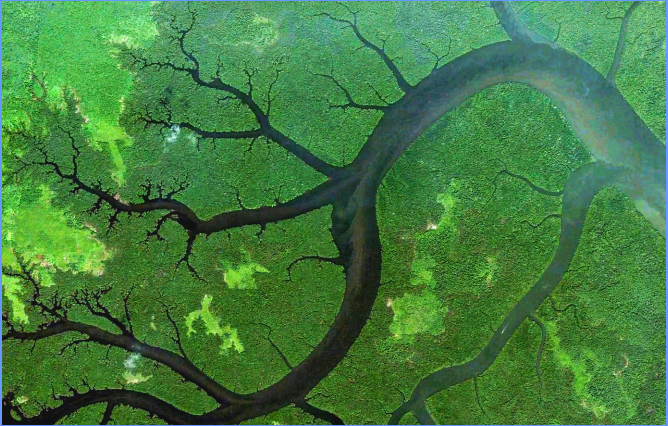
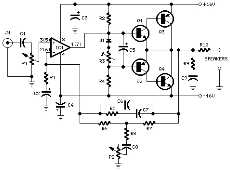
The Mathematics of Networks

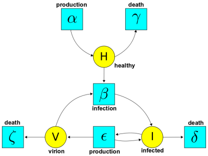


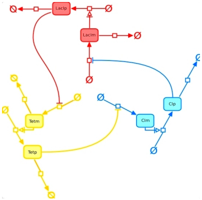
John Baez, U. C. Riverside

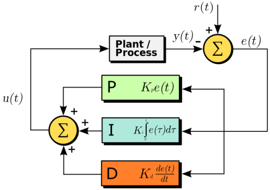
5 April 2018

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





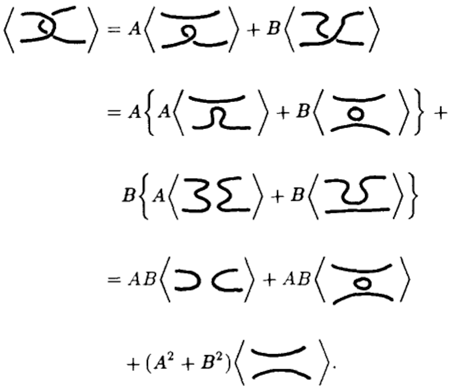




We need a good general theory of these.

Categories must be part of the solution. This became clear in the 1980s, when knot theory met particle physics:

Proof. (a)



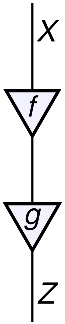
Part (b) is left for the reader.

Categories are great for describing processes of all kinds.

A category has **morphisms** $f \colon X \to Y$, which we can draw like this:

The input and output are called **objects**.

In a category we can **compose** morphisms $f\colon X \to Y$ and $g\colon Y \to Z$ to get a morphism $gf\colon X \to Z$, which we can draw like this:



We demand that some laws hold:

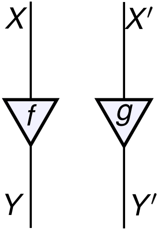
Definition. A **category** consists of:

- a collection of **objects**
- for any objects X and Y, a set of **morphisms** $f \colon X \to Y$

such that:

- given morphisms $f \colon X \to Y$ and $g \colon Y \to Z$ there is a **composite** morphism $gf \colon X \to Z$
- the **associative law** holds: $(hg)f = h(gf)$
- every object X has an **identity** morphism $1_X \colon X \to X$
- the **unit laws** hold: if $f \colon X \to Y$ then $1_Y f = f = f 1_X$.

In a **monoidal** category, we can also **tensor** morphisms $f\colon X \to Y$ and $g\colon X' \to Y'$ to get a morphism $f \otimes g\colon X \otimes X' \to Y \otimes Y'$, which we can draw like this:

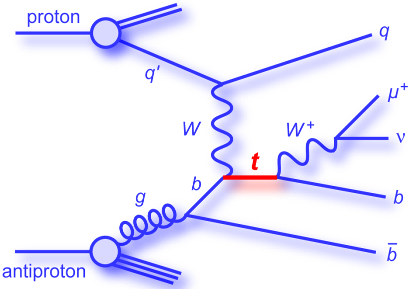


Monoidal categories must obey some additional laws, which are easy to find online.

In short: *composition* and *tensoring* let us assemble systems with inputs and outputs by putting together smaller systems *in series* and *in parallel*.

In the 1980s, particle physicists realized that any quantum field theory specifies a monoidal category!

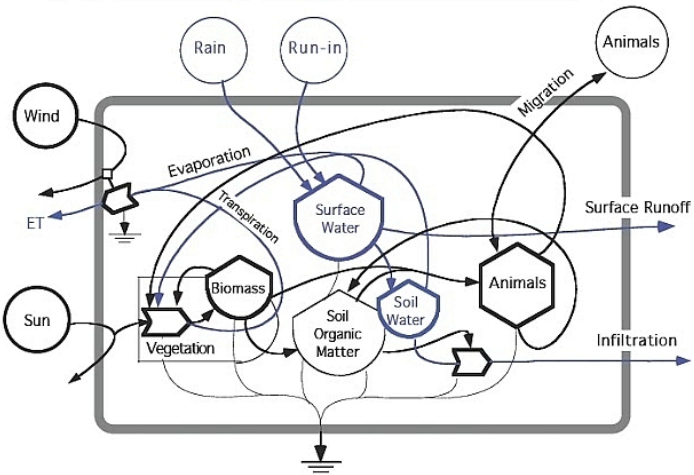
Feynman diagrams are just pictures of morphisms in such categories.



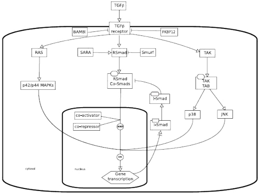
But why should particle physicists have all the fun? How about other subjects?

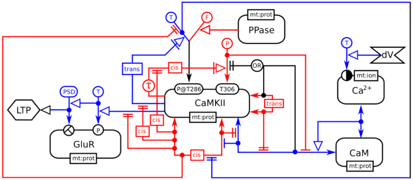
After all, this is the century of complex systems.

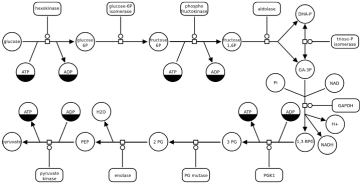
Back in the 1950's, Howard Odum introduced an Energy Systems Language for ecology:



Nowadays, biologists use *many kinds* of diagram languages.

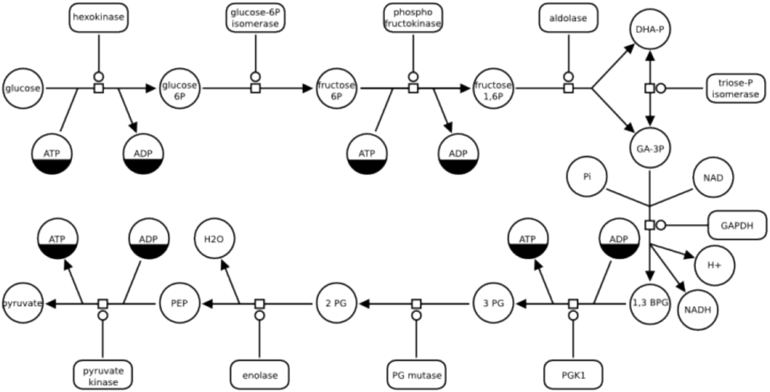




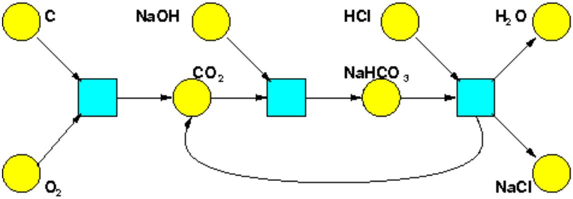


The Systems Biology Graphical Notation project is trying to standardize these.

For example, process diagrams show how entities interact and change from one type to another:



The simplest process diagrams are Petri nets. They were invented in 1939 for the purposes of chemistry:



as an alternative to the more familiar reaction networks:

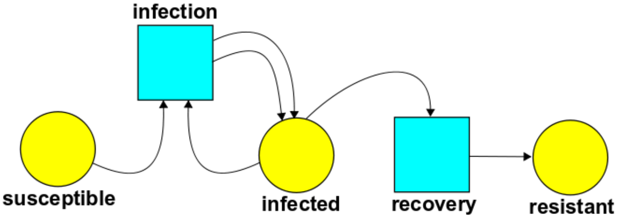
$$C + O_2 \to CO_2$$

$$CO_2 + NaOH \to NaHCO_3$$

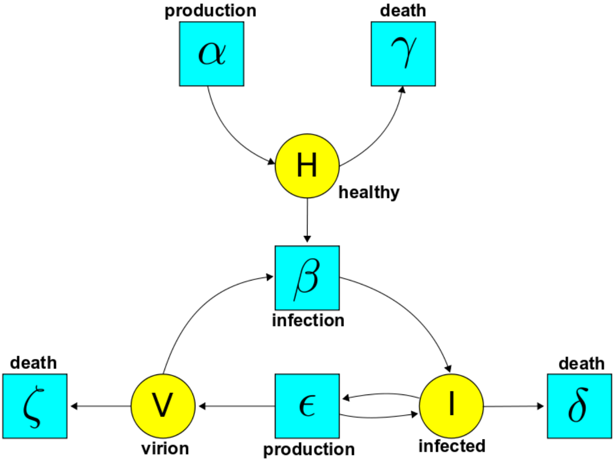
$$NaHCO_3 + HCl \to H_2O + NaCl + CO_2$$

In the 1970s, Petri nets were adopted by computer scientists as a model of ‘concurrency’.

They also serve as models of infectious disease:



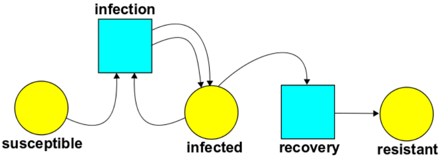
... or models of virus reproduction:



Definition. A **Petri net** consists of a set S of **species** and a set T of **transitions**, together with a functions

$$i\colon S \times T \to \mathbb{N}, \qquad o\colon S \times T \to \mathbb{N}$$

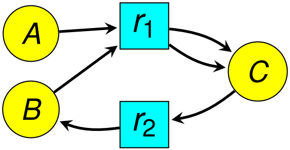
saying how many things of each species appear in the **input** and **output** of each transition.



$$S = \{\text{susceptible, infected, resistant}\}$$
$$T = \{\text{infection, recovery}\}$$

i(infected, infection) = 1, o(infected,infection) = 2

In a Petri net **with rates**, each transition $\tau \in T$ is assigned a **rate constant** $r(\tau) > 0$. We can then write down a **rate equation** describing dynamics. For example:



gives

$$
\begin{aligned}
\frac{dA}{dt} &= -r_1\, AB \\
\frac{dB}{dt} &= -r_1\, AB + r_2\, C \\
\frac{dC}{dt} &= 2r_1\, AB - r_2\, C
\end{aligned}
$$

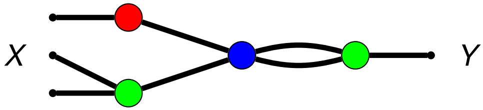
However, all these traditional Petri net models described *closed* systems.

Monoidal categories let us study *open* systems, where:

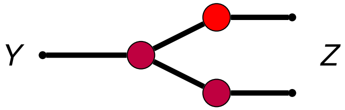
- entities can flow in or out of the system
- we can combine systems to form larger systems — by *composition* and *tensoring*.

The reason is that we can define categories where the morphisms are *networks*, and we compose them by gluing them together to form larger networks!

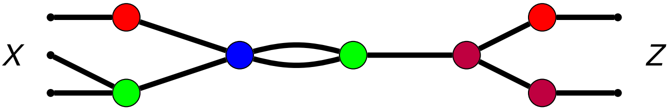
For example, there's a category where a morphism $f\colon X \to Y$ is a graph with colored nodes:



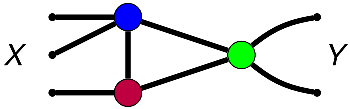
and given a morphism $g\colon Y \to Z$:



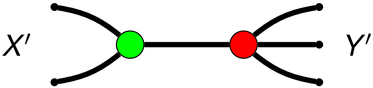
we form their composite by putting them in series:



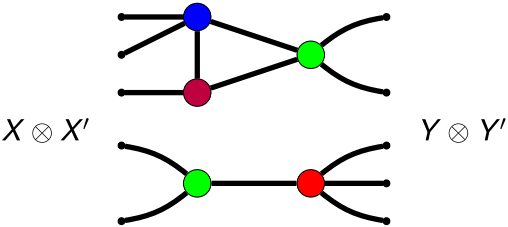
This category is monoidal. To tensor this morphism:



and this morphism:

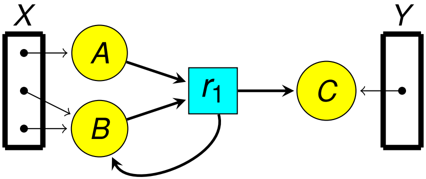


we put them in parallel:

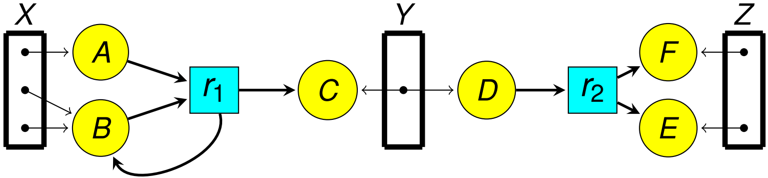


There are many variants! We can make a monoidal category of **open Petri nets with rates**, called **Petri**, where:

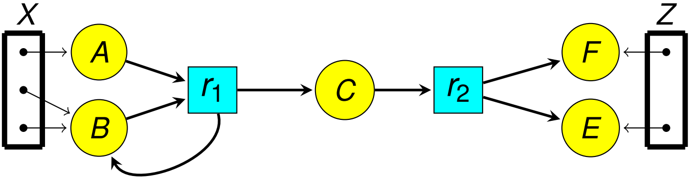
- An object is a finite set.
- A morphism $f\colon X \to Y$ is a Petri net with rates together with functions from X and Y to its set of species:



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

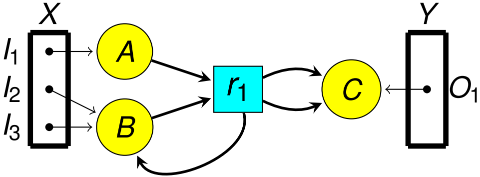


we put them in series, identifying output species of f with input species of g:



- To tensor morphisms, we put them in parallel.

Any open Petri net $f\colon X \to Y$ gives rise to an **open dynamical system** involving flows in or out, which can be arbitrary functions of time:



$$\frac{dA}{dt} = -r_1\, AB + I_1(t)$$

$$\frac{dB}{dt} = I_2(t) + I_3(t)$$

$$\frac{dC}{dt} = 2r_1\, AB - O_1(t)$$

Given an open Petri net with rates $f\colon X \to Y$, let's call its corresponding open dynamical system $\square(f)$.

There is a monoidal category **Dynam** where morphisms are open dynamical systems. Blake Pollard and I showed that

$$\square(gf) = \square(g)\square(f)$$

and

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

In other words, $\square$ preserves composition and tensoring!

One summarizes this by saying that

$$\square\colon \mathbf{Petri} \to \mathbf{Dynam}$$

is a monoidal functor.

The dream: each different kind of network or open system should be a morphism in a different monoidal category.

$$
\begin{tikzcd}
\mathbf{ResCirc} \arrow[r] \arrow[d] & \mathbf{Markov} \arrow[r] \arrow[ddl] & \mathbf{Petri} \arrow[d, "\square"] \\
\mathbf{Circ} \arrow[d] & & \mathbf{Dynam} \arrow[d, "\text{steady states}"] \\
\mathbf{SigFlow} \arrow[r, "\blacksquare"'] & \mathbf{LinRel}(\mathbb{C}(z)) \arrow[r, "\text{steady states}"'] & \mathbf{Rel}
\end{tikzcd}
$$

For more on category theory applied to networks:

- Brendan Fong, *The Algebra of Open and Interconnected Systems*.
- John Baez and Brendan Fong, A compositional framework for passive linear networks.

For more on Petri nets and reaction networks:

- John Baez and Jacob Biamonte, *Quantum Techniques for Stochastic Mechanics*.
- John Baez and Blake Pollard, A compositional framework for reaction networks.
- Blake Pollard, *Open Markov Processes and Reaction Networks*.

Also join my free online course on applied category theory on the Azimuth Forum!