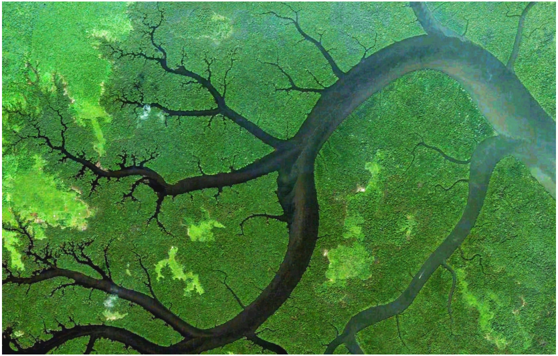
NETWORK THEORY

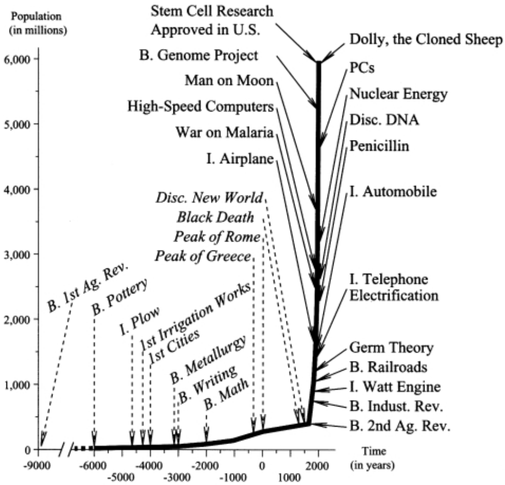


John Baez

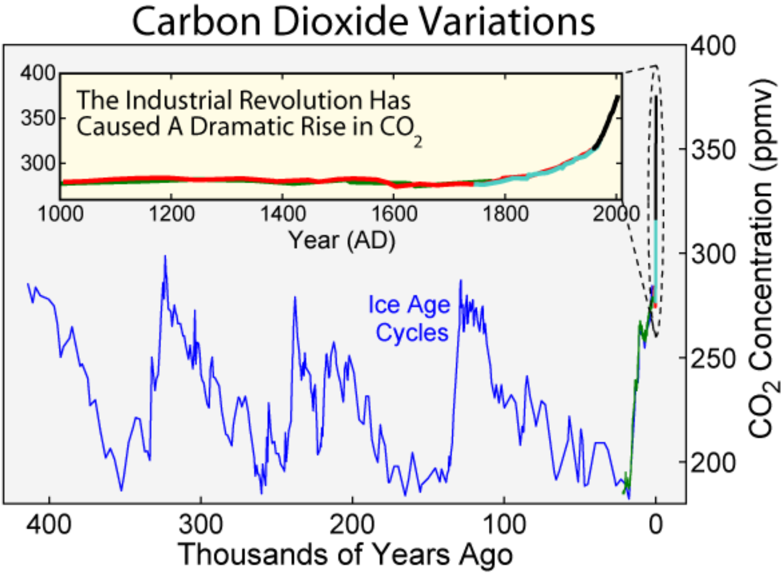
Dagstuhl

2 May 2014

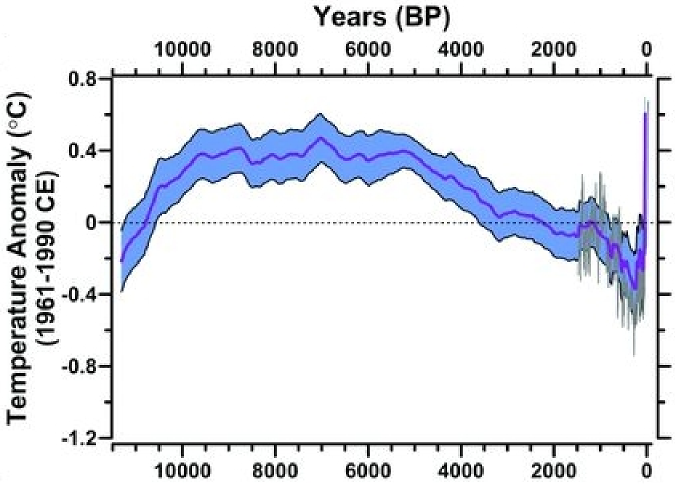
We have left the Holocene and entered a new epoch, the Anthropocene, when the biosphere is rapidly changing due to human activities.



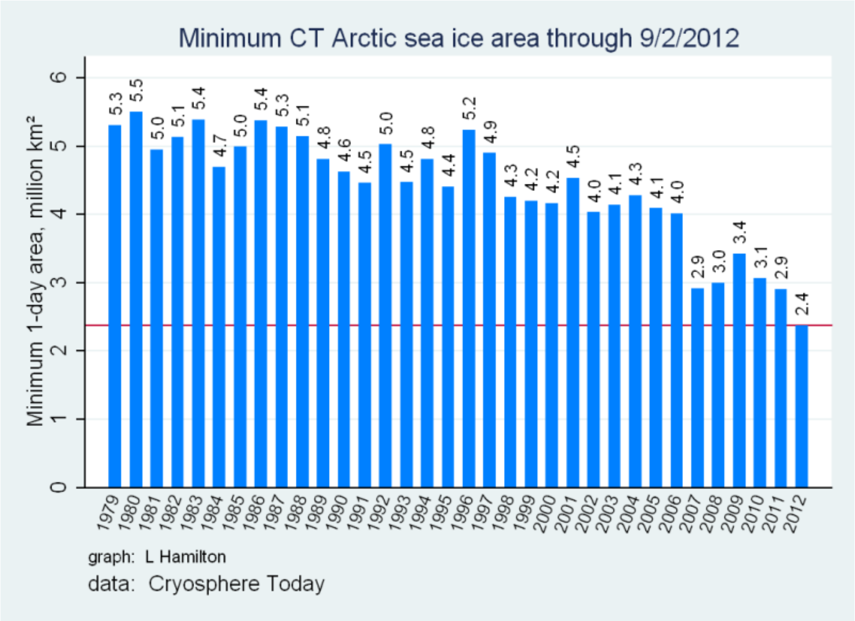
- About 1/4 of all chemical energy produced by plants is now used by humans.
- Humans now take more nitrogen from the atmosphere and convert it into nitrates than all other processes combined.
- 8-9 times as much phosphorus is flowing into oceans than the natural background rate.
- The rate of species going extinct is 100-1000 times the usual background rate.
- And then there's global warming....



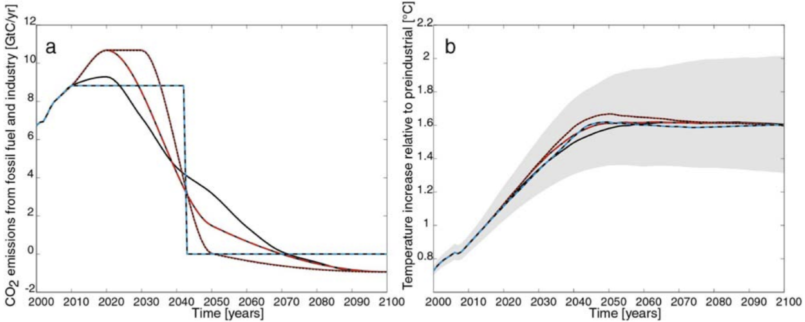
Antarctic ice cores and other data — Global Warming Art



Reconstruction of temperature from 73 different records — Marcott *et al.*


Minimum CT Arctic sea ice area through 9/2/2012
Minimum 1-day area, million km²
0
1
2
3
4
5
6
5.3
5.5
5.0
5.1
5.4
4.7
5.0
5.4
5.3
5.1
4.8
4.6
4.5
5.0
4.5
4.8
4.4
5.2
4.9
4.3
4.2
4.2
4.5
4.0
4.1
4.3
4.1
4.0
2.9
3.0
3.4
3.1
2.9
2.4
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
graph: L Hamilton
data: Cryosphere Today

According to the 2014 IPCC report on climate change, to surely stay below 2 °C of warming, we need a *more than 100% reduction in carbon emissions*...

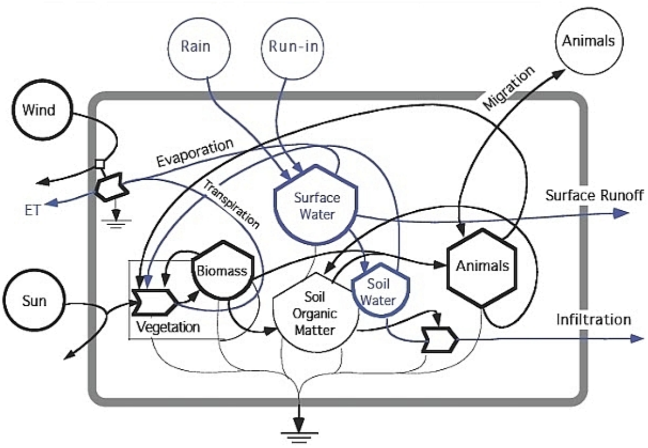


...unless we completely stop carbon emissions by 2040.

So, we can expect that in this century, scientists, engineers and mathematicians will be increasingly focused on *biology*, *ecology* and *complex networked systems* — just as the last century was dominated by physics.

What can category theory contribute?

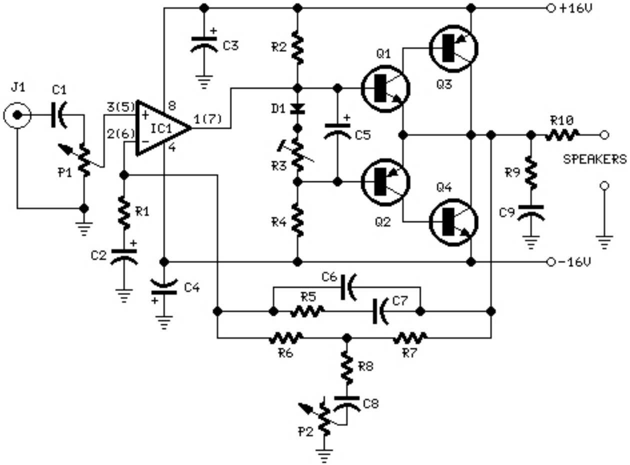
To understand ecosystems, ultimately will be to understand networks. — B. C. Patten and M. Witkamp



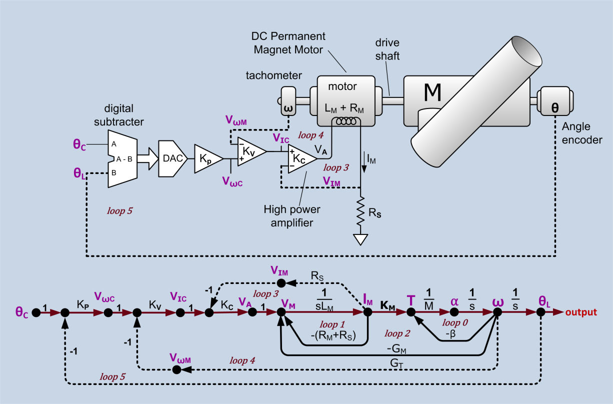
In the 1950's, Howard Odum introduced an Energy Systems Language to describe ecological networks.

Engineers, chemists, biologists and others now use *many* diagram languages to describe networks.

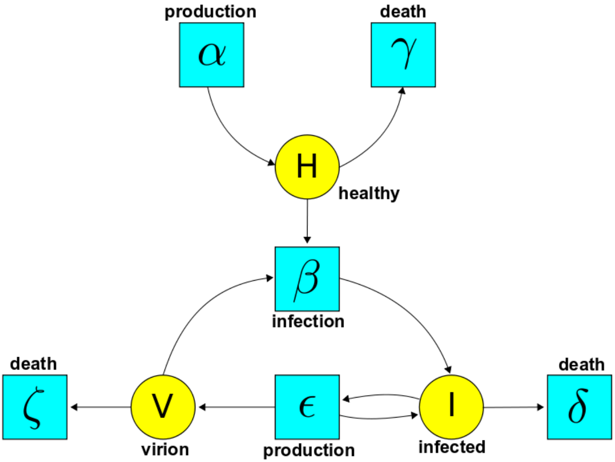
For example, electrical engineers use circuit diagrams:



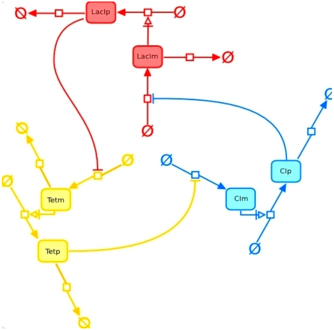
Control theorists use signal-flow graphs:



Stochastic Petri nets are used in chemistry, evolutionary game theory and epidemiology:

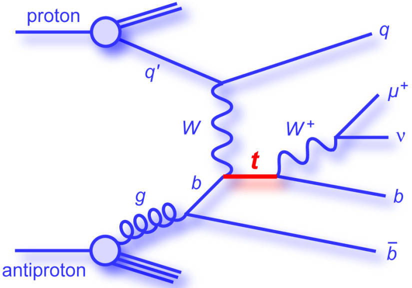


Systems Biology Graphical Notation has 3 diagram languages for biological networks. For example, 'process description language' generalizes stochastic Petri nets:



We need a good mathematical theory of *all* diagram languages!

Category theorists already treat 'string diagrams' as a syntax for morphisms in symmetric monoidal categories:



Physicists are starting to explicitly use 'functorial semantics':

$$F\colon \mathbf{Syntax} \to \mathbf{Semantics}$$

where F is a symmetric monoidal functor.

But we need to extend this idea from the rarefied world of particle physics to humbler but more practical applications!

Where to start? Three easy places:

1. electrical circuit diagrams and signal flow graphs
2. stochastic Petric nets ($\simeq$ chemical reaction networks) and Feynman diagrams
3. entropy, information and Bayesian networks

All these can be seen as 'warmup exercises' for biology and ecology.

Let's look at item 1.

Start with the simplest thing: circuits made of linear resistors!
Here Brendan Fong has constructed a symmetric monoidal functor

$$F\colon \mathbf{ResCirc} \to \mathbf{FinRel}_{\mathbb{R}}$$

where:

- **ResCirc** is a category with circuits made of resistors as morphisms
- $\mathbf{FinRel}_{\mathbb{R}}$ has finite-dimensional real vector spaces as objects and linear relations as morphisms — with $\oplus$ as tensor product!
- F sends any circuit with m input wires and n output wires to the linear relation between:
  - the input voltages and currents $(V, I) \in \mathbb{R}^{2m}$

  and
  - the output voltages and currents $(V', I') \in \mathbb{R}^{2n}$.

F treats the circuit as a 'black box', forgetting its internal structure and only remembering *what it does*.

We can generalize this in many ways:

- Circuits made of linear resistors, inductors, and capacitors:

$$F \colon \mathbf{LinCirc} \to \mathbf{LinRel}_{\mathbb{C}(z)}$$

  Now we get linear relations between finite-dimensional vector spaces over $\mathbb{C}(z)$: the field of complex rational functions in one variable z, which has the meaning of *differentiation*.

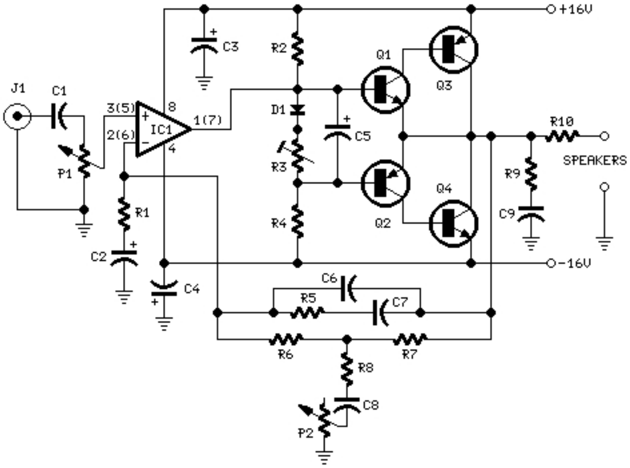
- Circuits that also include batteries and current sources:

$$F \colon \mathbf{AffCirc} \to \mathbf{AffRel}_{\mathbb{C}(z)}$$

  Now instead of *linear* relations between voltage and current we get more general *affine* ones.

- Nonlinear circuits — much more complicated and interesting! These need lots of work.

In the end we will see this as a morphism in an interesting symmetric monoidal category:

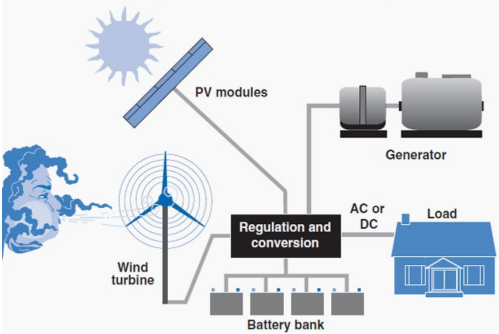


Why are electrical circuits worth so much study?

The mathematics governing them is *isomorphic* to that governing many other field of engineering!

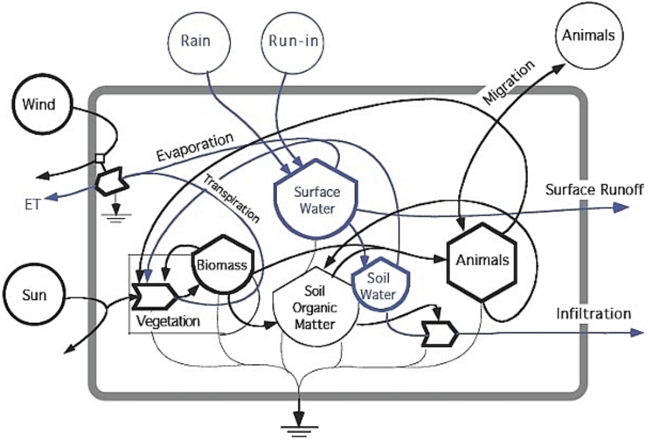
Field	q	$\dot{q}$	p	$\dot{p}$
Mechanics	position	velocity	momentum	force
Electronics	charge	current	flux linkage	voltage
Hydraulics	volume	flow	pressure momentum	pressure
Thermodynamics	entropy	entropy flow	temperature momentum	temperature
Chemistry	moles	molar flow	chemical momentum	chemical potential

Engineers often study **hybrid** — or in computer science terminology, 'typed' — systems involving mechanical, electronic and other elements:



These can be described using multi-typed versions of the symmetric monoidal categories mentioned so far.

Odum's Energy Systems Language also fits into this framework:



Some problems category theorists can solve. Given any 'syntax-to-semantics functor', e.g.

$$F \colon \mathbf{ResCirc} \to \mathbf{LinRel}_{\mathbb{R}}$$

it is interesting to ask:

- What morphisms are in the image of F? That is: what 'behaviors' can be 'realized'?

  (In this example, Fong has shown the answer is 'Dirichlet relations between symplectic vector spaces $\mathbb{R}^{2n}$'.)

- When does F map two morphisms $f, g \colon x \to y$ to the same morphism? Can we find a sufficient set of 'rewrite rules' that let us rewrite f and get g whenever $F(g) = F(f)$?

  (In this example, there is a nice answer for *planar* circuits made of resistors — see de Verdière–Gitler–Vertigan.)

- For any network, can we find a 'simplest' one that realizes the same behavior?

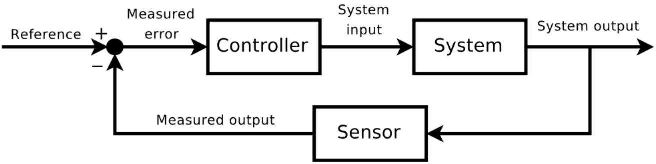
  For example, can we find a confluent terminating set of rewrite rules that let us rewrite any morphism $f\colon x \to y$ to a 'normal form': a specific choice of $g\colon x \to y$ with $F(g) = F(f)$?

  (For planar circuits made of resistors see Alman–Lian–Tran.)

Some problems category theorists should learn about

Network theory brings new issues from applied mathematics to the table of category theory!

For example, control theorists want to 'control' a system:



This involves the concepts of 'observability', 'controllability' and 'stability'.

- An electrical circuit is **observable** if by looking at the currents and voltages on output wires, we can determine those on the other wires.

- It is **controllable** if by controlling the currents and voltages on input wires, we can make those on the other wires be whatever we want.

These are 'dual' in the categorical sense. Observability says something is an *mono*. Controllability says something is an *epi*... but this needs clarification!

- A linear circuit is **stable** if bounded inputs produce bounded outputs.

Control theory is very interested in making a circuit 'stable' by composing and tensoring it with other circuits. For linear circuits, stability studied is using complex analysis. So, we should think about poles, etc. for morphisms in $\mathbf{LinRel}_{\mathbb{C}(z)}$.

Everything becomes harder and more interesting for nonlinear systems... like abrupt climate change!

The role of higher categories

Since networks are not just *processes* but also *things*, there are morphisms *between* networks! Examples include:

- symmetries of networks
- rewrites for simplifying networks
- evolution of networks over time

So, we have symmetric monoidal *bicategories* where the morphisms are networks, but there are also 2-morphisms *between* these morphisms.

k-tuply monoidal n-categories

	$n = 0$	$n = 1$	$n = 2$
$k = 0$	sets	categories	bicategories
$k = 1$	monoids	monoidal categories	monoidal bicategories
$k = 2$	commutative monoids	braided monoidal categories	braided monoidal bicategories
$k = 3$	‘’	symmetric monoidal categories	sylleptic monoidal bicategories
$k = 4$	‘’	‘’	symmetric monoidal bicategories
$k = 5$	‘’	‘’	‘’

A network diagram often amounts to a labelled graph with some designated 'inputs' and 'outputs'. It is thus a **cospan** in some category of labelled graphs.

A **cospan** is a diagram shaped like this:

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

A **map of cospans** is a diagram like this:

$$\begin{tikzcd}
 & S \arrow[dd, "f"'] & \\
X \arrow[ur, "i"] \arrow[dr, "i'"'] & & Y \arrow[ul, "j"'] \arrow[dl, "j'"] \\
 & S' &
\end{tikzcd}$$

where the triangles commute.

Theorem (Alex Hoffnung and Mike Stay)

For any category C with finite colimits, there is a symmetric monoidal bicategory $\mathrm{Span}(C)$ with:

- *objects of C as its objects*
- *cospans in C morphisms*
- *maps of cospans in C as 2-morphisms*

Moreover this symmetric monoidal bicategory is compact closed.

It also has a dagger structure:

$$
\dagger :
\begin{tikzcd}
 & S & \\
X \arrow[ur, "i"] & & Y \arrow[ul, "j"']
\end{tikzcd}
\quad \mapsto \quad
\begin{tikzcd}
 & S & \\
Y \arrow[ur, "j"] & & X \arrow[ul, "i"']
\end{tikzcd}
$$

As a consequence, there's a symmetric monoidal bicategory where:

- objects are finite sets $X, Y, \ldots$
- morphisms $f\colon X \to Y$ are circuits made of linear resistors, inductors, and capacitors going from X to Y. More technically, these are cospans of labelled graphs:

$$\begin{tikzcd} & S & \\ X \arrow[ur] & & Y \arrow[ul] \end{tikzcd}$$

- 2-morphisms are symmetries of circuits. More technically, these are invertible maps of cospans:

$$\begin{tikzcd} & S \arrow[dd, "\wr"'] & \\ X \arrow[ur] \arrow[dr] & & Y \arrow[ul] \arrow[dl] \\ & S' & \end{tikzcd}$$

Let us call this symmetric monoidal bicategory $\widetilde{\mathbf{LinCirc}}$. There is a symmetric monoidal functor called 'decategorification'

$$\mathbf{Decat}\colon \widetilde{\mathbf{LinCirc}} \to \mathbf{LinCirc}$$

sending

- each object to itself,
- each morphism to its isomorphism class,
- each 2-morphism to an identity 2-morphism.

where as usual, we treat a category as a bicategory with only identity 2-morphisms.

In plain English:

$$\mathbf{Decat}\colon \widetilde{\mathbf{LinCirc}} \to \mathbf{LinCirc}$$

identifies any two circuits related by a symmetry and then discards the symmetries.

So: for many purposes we can work with the category **LinCirc** where we pretend isomorphic circuits are equal...

...but when we want, we can work with the bicategory $\widetilde{\mathbf{LinCirc}}$ where we admit they are merely isomorphic!

There's a commutative square

$$\begin{tikzcd} \widetilde{\mathbf{LinCirc}} \arrow[r, "\mathbf{Decat}"] \arrow[d] & \mathbf{LinCirc} \arrow[d] \\ \mathbf{SigFlow} \arrow[r, "\mathbf{Decat}"] & \mathbf{LinRel}_{\mathbb{C}[z]} \end{tikzcd}$$

Here the horizontal arrows are decategorification functors, the vertical arrows are 'syntax-to-semantics' functors, and **SigFlow** is the symmetric monoidal bicategory where

- objects are finite sets
- morphisms are signal-flow graphs as in control theory,
- 2-morphisms are symmetries of signal-flow graphs.

There's a lot to do! You can see references by clicking on words in blue in this talk, available here:

http://tinyurl.com/networks-dagstuhl