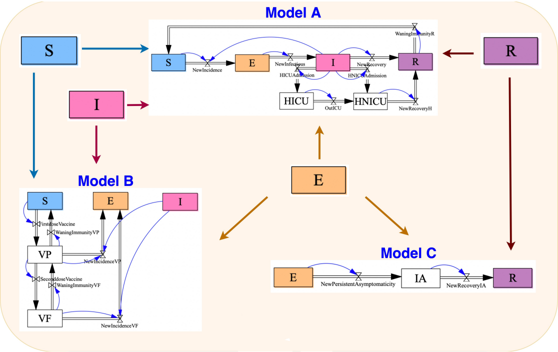
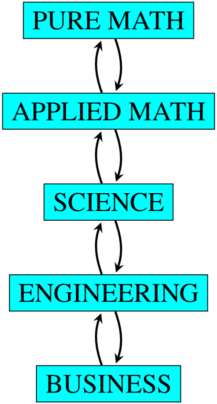
Applied Category Theory



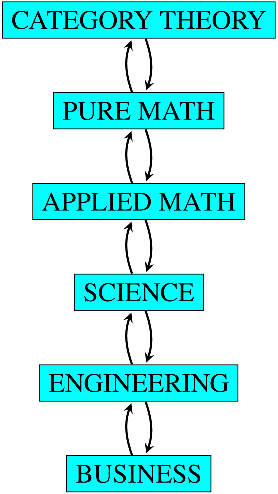
John Baez, U. C. Riverside

2023 June 6

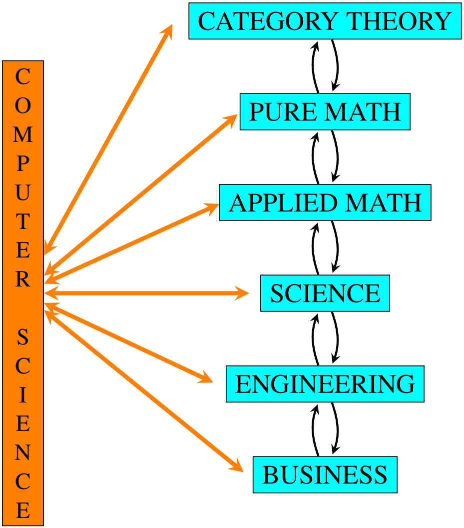
We often imagine information flowing beween pure math and its most concrete applications in steps.



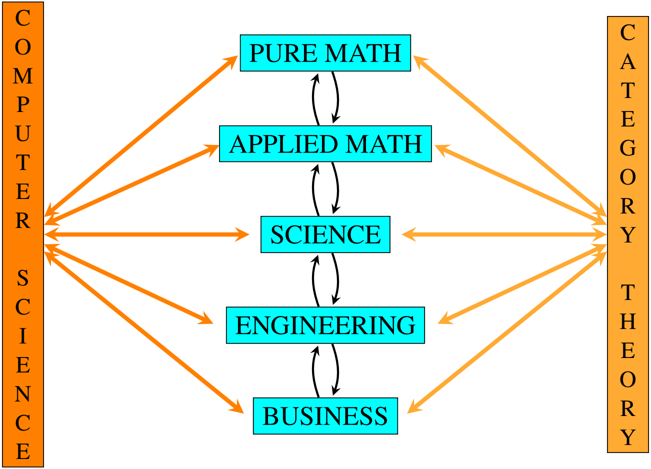
We often imagine information flowing beween pure math and its most concrete applications in steps. In this picture, category theory is the "purest of the pure".



Now computer science cuts across this traditional flow of information.



Now computer science cuts across this traditional flow of information. Maybe category theory can too?



In his 1963 thesis, Lawvere introduced “functorial semantics”, where there is a category C of syntactic expressions, and a choice of functors $F\colon \mathrm{C} \to \mathrm{D}$ sending expressions to their meanings.

This became important in computer science. For example a nice programming language may give a category C where objects are “data types" $x, y, \ldots$ and a morphism $f\colon x \to y$ is a program that accepts data of type x as input and outputs data of type y.

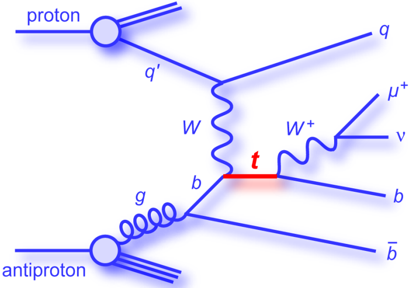
A functor $F\colon \mathrm{C} \to \mathrm{Set}$ then maps each data type x to a set and each program $f\colon x \to y$ to a function, obeying

$$F(f \circ g) = F(f) \circ F(g)$$

$$F(1_x) = 1_{F(x)}$$

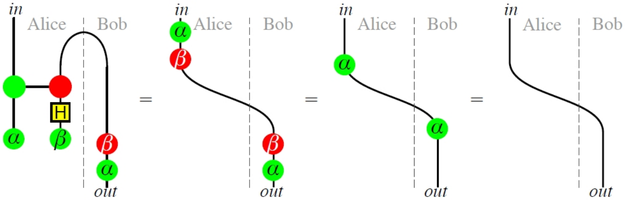
In the 1980s, particle physicists realized that any quantum field theory specifies a category C where objects are collections of particles and morphisms are ways for particles to interact and turn into other particles.

Feynman diagrams are pictures of morphisms in such categories.

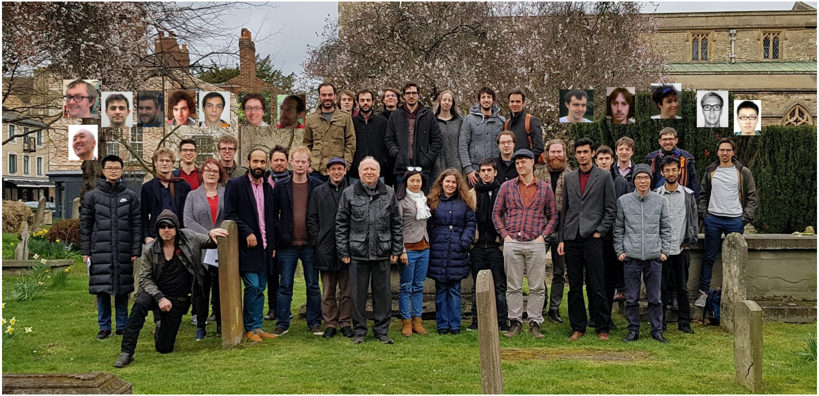


A functor $F \colon \mathrm{C} \to \mathrm{Hilb}$ turns a Feynman diagram into a linear operator between Hilbert spaces.

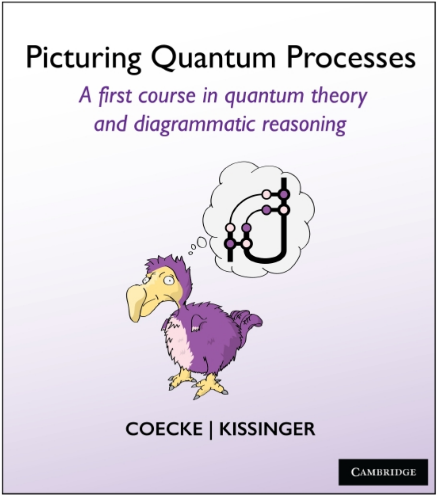
In 2004 Samson Abramsky and Bob Coecke showed that “quantum teleportation” and other forms of quantum information processing could be understood using category theory and Feynman-like diagrams.



Later they built a huge group at Oxford working on categories, the foundations of quantum physics, and quantum computation.

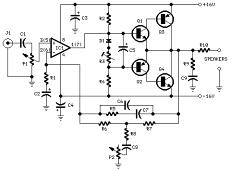


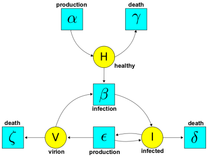
Coecke and coauthors developed a diagrammatic method for explaining quantum physics, relying on categories:

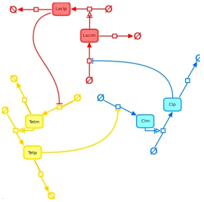


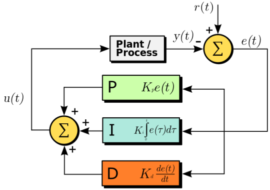
In 2021, Coecke left Oxford University and became the chief scientist of Quantinuum, a company that uses categorical methods in quantum computing and also natural language processing.

But functorial semantics is very general. In *many* areas of science and engineering, people use a syntax where morphisms are diagrams, and a semantics where some functor maps these to systems of differential equations, etc.

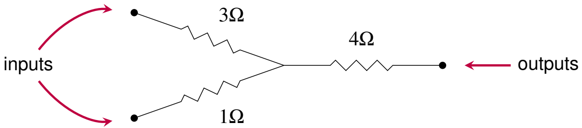






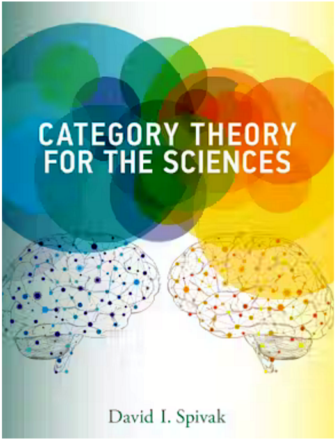


Around 2012, I asked Brendan Fong to create and study a category having “open electrical circuits” as morphisms:

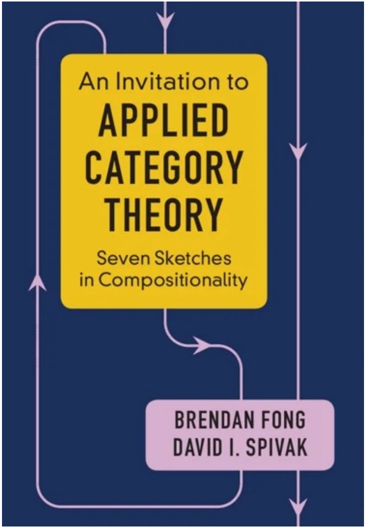


He invented the theory of “decorated cospans” to do this, and finished his thesis on it in 2016.

In 2016 Fong went to MIT to work as a postdoc with David Spivak. Spivak had helped create a language for databases, Functorial Query Language, which is now the basis of a company, Conexus AI. In 2014 he wrote a book explaining some of his ideas.



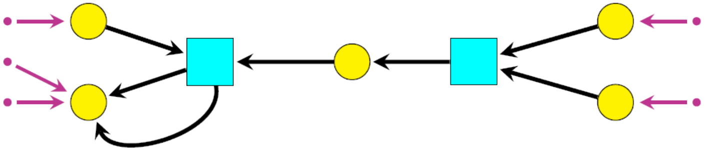
In 2018, Fong and Spivak wrote a textbook on applied category theory and taught a course on it, both freely available online.



In 2018 Kenny Courser and I came up with "structured cospans", a simpler alternative to decorated cospans. In 2020, Christina Vasilakopolou helped us understand how they're related.

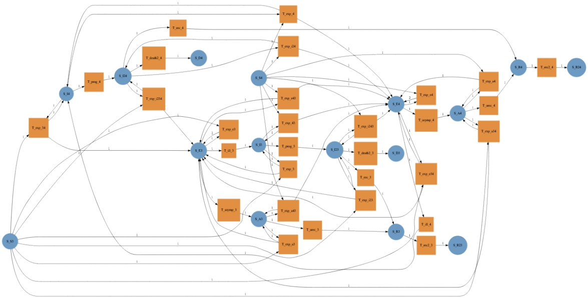
By now, structured and decorated cospans have been used to study open systems of many kinds:

- open graphs
- open Petri nets
- open chemical reaction with rates
- open electrical circuits
- open Markov processes
- open dynamical systems



In 2019, James Fairbanks and Evan Patterson began developing AlgebraicJulia, a system for high-performance scientific computing that lets you program using constructs from category theory: categories, functors, natural transformations, presheaves, operads, etc.

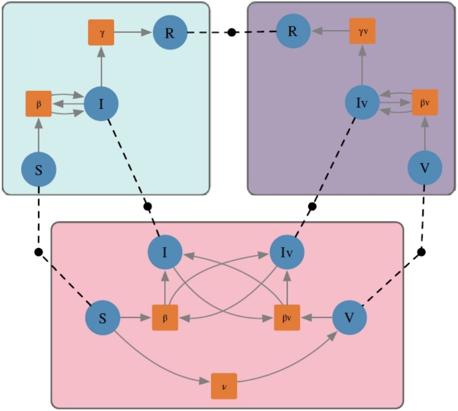
In 2020, Patterson implemented structured cospans in AlgebraicJulia. Using this system to compose open Petri nets, he and Micah Halter reconstructed some of a COVID-19 model used by the UK government.



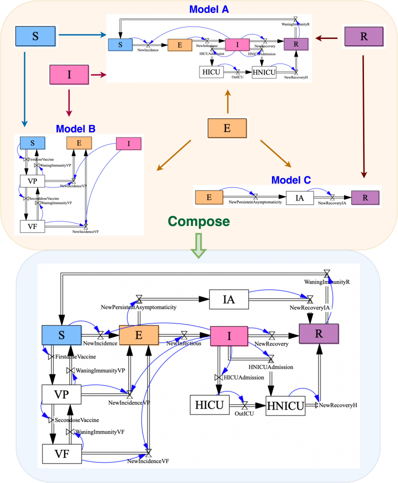
This attracted the attention of some epidemiologists.

In 2021, Brendan Fong and David Spivak founded the Topos Institute, an institute for applied category theory in Berkeley. They hired Evan Patterson and Sophie Libkind, and others.

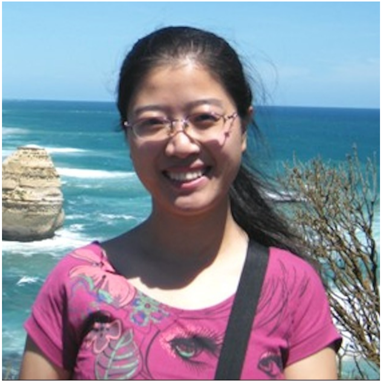
In 2022, Sophie Libkind, Andrew Baas, Micah Halter, Evan Patterson and James Fairbanks used AlgebraicJulia to build models of epidemiology using open Petri nets with rates:



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



There is a community of epidemiologists who use stock-flow diagrams to model the spread of disease. This includes my coauthors Nathaniel Osgood and Xiaoyan Li, who did COVID modeling for the government of Canada.



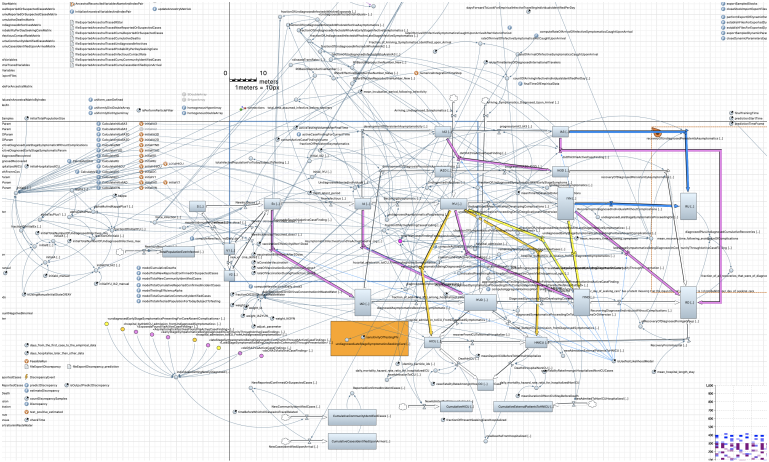
In “community-based modeling” à la Hovmand, stock-flow diagrams make models understandable to nonexperts, who can then join in building models.

Most stock-flow modeling is done using software called AnyLogic. It's powerful, but it has several big problems:

- It has no support for *composing* models: that is, taking several smaller models and putting them together to form a larger model.
- It doesn't separate *syntax* from *semantics*.
- It has no support for *"stratifying"* models: that is, taking a model and splitting one stock into several stocks (e.g. age groups).
- It has no support for *collaboratively* building models.
- It is not *free* and not *open-source*!

Our new work aims to fix all these problems.

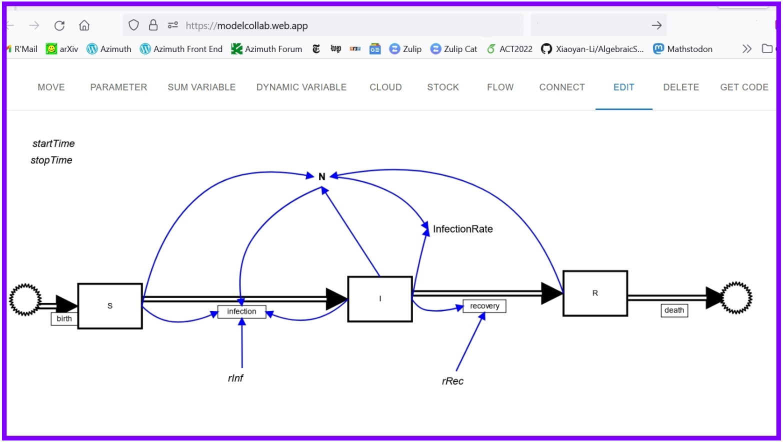
The ability to *compose* models is crucial because realistic models are complicated and built out of many smaller parts. Here is Osgood and Li's COVID model used by the government of Canada:



We have now created software that supports compositional modeling with stock-flow diagrams. It's called StockFlow and it's available for free on GitHub. It lets you:

- create and compose open stock-flow diagrams
- "stratify" open stock-flow diagrams
- apply several different semantics functors to a stock-flow diagram, one turning it into a system of differential equations
- *solve* the differential equations.

Nathaniel Osgood and Eric Redekopp have now made a graphical user interface for StockFlow. This software, called ModelCollab, runs in multiple browsers, so teams can collaborate to build stock-flow diagrams.



The next step is to train epidemiologists to use ModelCollab. Luckily:

- Nathaniel Osgood runs regular “bootcamps” on epidemiological modeling.
- No understanding of category theory or the programming language Julia is requred to use ModelCollab!
- ModelCollab is free, while AnyLogic is not.

Still, it will take work. For applied math to be *truly* applied takes many steps.

Applied category is interesting in many ways.

- It requires new forms of communication and collaboration. One is the annual ACT conference, which started in 2018. Another is the journal *Compositionality*. Another is the Category Theory Community Server.
- It raises ethical issues category theorists are not used to. For example: the biggest funder of research on applied category theory is currently the US military. Also, applied category theory is being used for AI. Will it merely make the richest and most powerful still richer and more powerful, or can it do something truly new?
- It revitalizes the old question: which mathematical structures are best for describing and designing systems? It opens some possibilities for radical progress here.