All concepts are $\mathbb{C}\mathbf{at}^\sharp$

Owen Lynch   Brandon Shapiro   **David I. Spivak**

TOPOS INSTITUTE

Applied Category Theory 2023
2023 / 08 / 04

Outline

Why am I here?

For 15 years, I've wanted CT to help humanity make sense of its world.

- Data migration: differently-organized systems exchanging info.
- Operadic compositionality: new things built from arrangements of old.
- Interacting dynamical systems: how collectives act in concert.

Why math?

- I think of mathematical fields as *accounting systems*.
- We account for quantities, likelihoods, physics observ'ns, reasoning...
- ...using arithmetic, probability, Hilbert spaces, logic.
- Math's universality and fine-tuned language lead to impressive coord'n.

Why category theory?

- CT is even more fine-tuned. The language and principles are elegant.
- As constructive, it is more amenable to tool-building, applications.
- It's a microcosm of math: sitting within it and reflecting its structure.

Almost the same story repeats at another level, within CT.

All concepts?

Categories Work has a section titled "All concepts are Kan extensions."

- This is an exaggeration: Lots of CT ideas are not Kan extensions.
- But Kan ext'ns are so far-reaching, e.g. adjoints, Yoneda, (co)limits,...
- ...that the exaggeration is worthwhile.

The double category $\mathbb{C}\mathbf{at}^\sharp$ is similarly far-reaching.

- It includes categories, (co)functors, profunctors, and natural transf'ns.
- It includes all copresheaf categories, elements, and pra-functors.
- It internally constructs nerves of categories and higher categories.
- It includes $\mathbb{P}\mathbf{oly}_{\mathcal{E}}$ for any category $\mathcal{E}$ with pullbacks, e.g. multivariate.
- It models dynamic organizational structures ($\mathbb{O}\mathbf{rg}$) as in deep learning.
- And many other app'ns (effect handlers, rewriting, data migration, etc)

Elegant, applicable, and far-reaching, it's an important part of ACT.

Plan for the talk

In this talk I will:

- Introduce **Poly** and $\mathbb{C}\mathbf{at}^\sharp$,
- Show three homes for categories in $\mathbb{C}\mathbf{at}^\sharp$,
- Explain how multivariate polynomials and $\mathbb{P}\mathbf{oly}_\mathcal{E}$ fit in,
- ~~Discuss nerves of categories and higher categories~~ oops, this is ACT,
- Recall dynamic arrangements and show how they embed, and
- Conclude.

Outline

What is Poly?

There are many equivalent ways to get **Poly**, e.g.

- The free completely distributive category ($\Pi\Sigma \to \Sigma\Pi$) on one object.
- The full subcat'y of functors **Set** $\to$ **Set** on coprod's of representables.
- The full subcat'y of functors **Set** $\to$ **Set** preserving connected limits.

Let's bring it down to earth.

- A representable functor is one of the form $X \mapsto X^A$. Denote it y^A.
- Coproducts of such things—objects of **Poly**—are denoted $\sum_{i:I} y^{A_i}$.
- Maps between these things are easy, by UP of coproducts and Yoneda:

$$\mathbf{Poly}\left(\sum_{i:I} y^{A_i}, \sum_{j:J} y^{B_j}\right) \cong \prod_{i:I}\sum_{j:J} \mathbf{Set}(B_j, A_i).$$

The category **Poly** has an unprecedented amount of structure.[1]

- All limits and colimits, left Kan ext'ns, three factorization systems.
- Infinitely many *monoidal closed* structures.
- Free monads, cofree comonads, and lawful interactions between them.

It has many applications in functional, imperative, automata programming.

[1] See arxiv.org/abs/2202.00534 for a compressed reference on **Poly**'s structure.

Polynomial comonads are categories

Polynomial functors can be composed; this operation is a monoidal product.

- Considering polynomials as objects, we write $p \triangleleft q$ rather than $p \circ q$.
- It's just like composing polynomials normally: $y^2 \triangleleft (y+1) \cong y^2+2y+1$.
- So one can ask: what are monoids and comonoids in $(\mathbf{Poly}, y, \triangleleft)$?
- As functors **Set** $\to$ **Set**, these are called poly'l monads and comonads.

Let's just work with comonads. How can you think about them?[2]

- Amazing fact: polynomial comonads are exactly categories!
- Morphisms between them are not functors; they're called *cofunctors*.
- A polynomial comonad is a tuple (c, ϵ, δ), where c : **Poly** and

$$\epsilon \colon c \to y \qquad \text{and} \qquad \delta \colon c \to c \triangleleft c$$

How do we think of this like a category? Let $\mathcal{C}$ be a category.

- For each object $A : \mathrm{Ob}(\mathcal{C})$, let $\mathcal{C}[A] := \sum_{B:\mathrm{Ob}(\mathcal{C})} \mathcal{C}(A, B)$, "maps out"
- Then the associated polynomial is $\sum_{A:\mathrm{Ob}(\mathcal{C})} y^{\mathcal{C}[A]}$.
- Counit ϵ supplies id's; comult δ supplies codomains and composites.

[2]These results are due to Ahman-Uustalu.

What is $\mathbb{C}\mathbf{at}^\sharp$?

In *Framed Bicategories*, Shulman defines the $\mathbb{M}\mathbf{od}$ construction.

- If a double cat'y $\mathbb{D}$ has nice local coequalizers, you can form $\mathbb{M}\mathbf{od}(\mathbb{D})$.
- Similarly, if $\mathbb{P}$ has nice local equalizers, you can form $\mathbb{C}\mathbf{omod}(\mathbb{P})$.
- Any monoidal cat'y is a vertically trivial double category.
- Let $\mathbb{P}$ be the one-object double cat'y associated to $(\mathbf{Poly}, y, \triangleleft)$.
- It has nice ($\triangleleft$-preserved) local equalizers: $e \to p \rightrightarrows q$.
- So we can form $\mathbb{C}\mathbf{omod}(\mathbb{P})$. I refer to this double category as $\mathbb{C}\mathbf{at}^\sharp$.

Why do I call it $\mathbb{C}\mathbf{at}^\sharp$?

- By Ahman-Uustalu, its objects are precisely all small categories.
- But verticals in $\mathbb{C}\mathbf{at}^\sharp$ are a little *sharp*; they are *co*functors.
- Garner[3] explained that its horizontals $c \triangleleft\!\!\longrightarrow\!\!\triangleleft\, d$ are very cool things.
- They're parametric right adjoint (pra) functors d-$\mathbf{Set} \to c$-$\mathbf{Set}$.
- These are exactly data migrations from d-databases to c-databases.
- They generalize profunctors: they're $\mathcal{C}$-indexed sums of profunctors.

[3] See Garner's HoTTEST video: `https://www.youtube.com/watch?v=tW6HYnqn6eI`

Outline

Three homes for categories

$\mathbb{C}\mathbf{at}^\sharp$ is the equipment of comonoids in the distributive completion of 1.

- Why do we care about distributive completions or comonoids?
- You might not guess this was cool, a priori.
- All you'd know is that it has a very short description in CT language.
- That's often a good sign, e.g.
  $$\mathbb{C}\mathbf{omod}(\mathbf{Set}, 1, \times) \cong \mathbb{S}\mathbf{pan} \quad \text{and} \quad \mathbb{M}\mathbf{od}(\mathbb{S}\mathbf{pan}) \cong \mathbb{P}\mathbf{rof}\mathbb{C}\mathbf{at}.$$
- But there's a lot more to say about $\mathbb{C}\mathbf{at}^\sharp = \mathbb{C}\mathbf{omod}(\mathbf{Poly}, y, \triangleleft)$.

Our first goal is to bring you some feeling of familiarity with $\mathbb{C}\mathbf{at}^\sharp$.

- We'll find three different homes for categories in $\mathbb{C}\mathbf{at}^\sharp$.
- First is the least familiar but most top of mind:...
- Categories are the comonoids in **Poly**, so they're the objects of $\mathbb{C}\mathbf{at}^\sharp$.
- You can find functors in this home, but tucked away, hardly relevant.
- Every functor $\mathcal{C} \to \mathcal{D}$ shows up as an adjunction $c \underset{\Pi_F}{\overset{\Delta_F}{\rightleftarrows}} d$
- They constitute the left class of a factorization system on left adjoints.

There are better homes if you want to hang out with ordinary cat'ies.

$\mathbb{S}$pan lives inside $\mathbb{C}$at$^\sharp$ as the linears

We never said what a horizontal morphism in $\mathbb{C}\mathbf{at}^\sharp$ is. It's a *bicomodule*.

$$c \triangleleft p \leftarrow p \rightarrow p \triangleleft d \qquad \text{satisfying laws w.r.t. } \epsilon\colon c \rightarrow y\text{, etc.}$$

I'm still astounded that these are precisely prafunctors d-**Set** $\rightarrow$ c-**Set**.

- A single poly'l (plus two lawful maps) governs the data migration.
- Example: if $d = 0$, one can show that p must be a set, $p = Py^0 = P$.
- Bicomodules $c \overset{P}{\triangleleft\!\!-\!\!-\!\!\triangleleft} 0$ can be identified with functors $c \rightarrow$ **Set**.

So what would you get if you only looked at linear polynomials?

- Take as objects only linear comonoids $c = Cy$ for some C : **Set**.
- Take as verticals all maps, and as horizontals linear bicomodules

$$Cy \overset{Py}{\triangleleft\!\!-\!\!-\!\!\triangleleft} Dy$$

- The result is exactly $\mathbb{S}\mathbf{pan} \cong \mathbb{C}\mathbf{omod}(\mathbf{LinPoly}, y, \triangleleft)$.

It's well-known that monads in $\mathbb{S}\mathbf{pan}$ are categories, $\mathbb{M}\mathbf{od}(\mathbb{S}\mathbf{pan}) \cong \mathbb{C}\mathbf{at}$.

- If you're new to this, it's worth thinking about/asking someone.
- Anyway, the second home: monads in the linear subcat'y of $\mathbb{C}\mathbf{at}^\sharp$.

The most familiar: path-algebras

While objects in $\mathbb{C}\mathbf{at}^\sharp$ are cat'ies, they act like copresheaf cat'ies.

- We said that the cat'y of bicomodules $c \triangleleft\!\!\longrightarrow\!\!\triangleleft\, 0$ is c-**Set**...
- ...and in general, bicomodules $c \triangleleft\!\!\longrightarrow\!\!\triangleleft\, d$ are praf'rs d-**Set** $\to$ c-**Set**.
- Can we find categories in terms of copresheaves?

Let $\mathcal{G} := \boxed{\bullet^E \rightrightarrows \bullet^V}$. The corresponding polynomial is $g := y^3 + y$.

- The cat'y of $\mathcal{G}$-sets, i.e. bicomodules $g \triangleleft\!\!\longrightarrow\!\!\triangleleft\, 0$, is g-**Set** $\cong$ **Grph**.
- A bicomodule $g \triangleleft\!\!\longrightarrow\!\!\triangleleft\, g$ is a prafunctor g-**Set** $\to$ g-**Set**.
- Prafunctors may be new to you, but they're a really nice, general class.
- Here's a good one: $g \overset{path}{\triangleleft\!\!\longrightarrow\!\!\triangleleft} g$. It sends a graph $g \overset{G}{\triangleleft\!\!\longrightarrow\!\!\triangleleft} 0$ to...
- ... $g \overset{path}{\triangleleft\!\!\longrightarrow\!\!\triangleleft} g \overset{G}{\triangleleft\!\!\longrightarrow\!\!\triangleleft} 0$, which is the graph of all paths in G.

It's well-known that *path* is a monad; its category of algebras is **Cat**!

- So categories are graphs G equipped with a map $path \triangleleft_g G \to G$...
- ...satisfying the monad algebra axioms.
- **Cat**'s home as the path-complete graphs is probably most familiar.

Outline

Common complaints about Poly

Everyone recognizes that **Poly** is overflowing with structure.

- Limits, colimits, infinitely many monoidal closed structures, etc.
- Have you ever heard of four monoidal structures interacting like this?

$$(p_1 \triangleleft p_2 \triangleleft p_3) \times (q_1 \triangleleft q_2 \triangleleft q_3) \to (p_1 \otimes q_1) \triangleleft (p_2 \times q_2) \triangleleft (p_3 + q_3)$$

- This map is actually surprisingly useful, but I digress.

But people naturally want more. Here are the two most common asks:

- "I want multivariate polynomials; you only care about univariate y."
- "I want polynomials in $\mathcal{E}$; you only care about **Set**."

Let's consider both of those at once.

- N. Gambino and J. Kock wrote a beautiful paper about polynomials.
- For any locally cartesian closed category $\mathcal{E}$, they define...
- ...an equipment $\mathbb{P}\mathbf{oly}_\mathcal{E}$ of multivariate polynomials in $\mathcal{E}$.

Multisorted polynomials over arbitrary $\mathcal{E}$

If $\mathcal{E}$ is a category with pullbacks, one can define a double category $\mathbb{P}\mathbf{oly}_\mathcal{E}$.

- Univariate polynomials are exponentiable maps $E \to B$ in $\mathcal{E}$.
- Multivariate polynomials are "bridge diagrams" in $\mathcal{E}$:

$$I \leftarrow E \to B \to J$$

- For example if $\mathcal{E} = \mathbf{Set}$ then this is J-many poly's in I-many variables.

Let's find $\mathbb{P}\mathbf{oly}_\mathcal{E}$ inside $\mathbb{C}\mathbf{at}^\sharp$.

- First, find any full dense subcategory $\mathcal{A}^{\mathsf{op}} \subseteq \mathcal{E}$, e.g. $\mathcal{A}^{\mathsf{op}} = \mathcal{E}$.
- The cat'y of univariate polys in $\mathcal{E}$ embeds fully faithfully...
- ...and strong monoidally into the bicomodule category $\mathbb{C}\mathbf{at}^\sharp(\mathcal{A}, \mathcal{A})$.
- The multivariate double category $\mathbb{P}\mathbf{oly}_\mathcal{E}$ embeds into $\mathbb{C}\mathbf{at}^\sharp$ by...
- ...sending $I : \mathcal{E}$ to the slice category $\mathcal{A}/I$ and a bridge diagram...
- ...as above to a certain bicomodule $\mathcal{A}/I \mathrel{\triangleright\!\!\longrightarrow\!\!\triangleright} \mathcal{A}/J$.

In particular, if you just want multivariate polynomials in **Set**:

- Note that $1 \subseteq \mathbf{Set}$ is dense. The double category $\mathbb{P}\mathbf{oly}_{\mathbf{Set}}$ is...
- ...the full sub double cat'y of $\mathbb{C}\mathbf{at}^\sharp$ spanned by the discrete categories.

Outline

Dynamic functions

Let's get to applications. People often refer to functions as machines.

- A function $f\colon A \to B$ takes in A's and spits out B's.
- It is automatic, deterministic, total, unchanging through use.

In real life, machines change as you use them.

- An over-used key on your keyboard might have a faded letter.
- Your shoes wear down according to how you walk.
- Similarly for your baseball glove, your brain, your home.
- Automatic, deterministic, total, but they *change based on usage*.

I want to call such a thing a *dynamic function* $A \to B$.

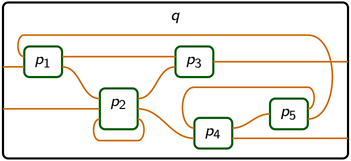
- They're modeled by *Mealy machines*, i.e...
- ...a set S of "states" and a function $f\colon S \times A \to B \times S$.

These are exactly $[Ay, By]$-coalgebras.

- $[p, q]$ is the inner hom for a monoidal structure denoted $\otimes$.
- We have $[Ay, By] \cong (By)^A$. So a coalgebra $S \to [Ay, By](S)$...
- ...is a function $S \to (BS)^A$, which curries to $S \times A \to B \times S$.

Dynamic arrangements

Everything above also works for **Poly** maps, e.g. wiring diagrams



$\varphi : [p_1 \otimes \cdots \otimes p_5, q](1)$

Mealy machines are $[Ay, By]$-coalgebras; what are $[p, q]$-coalgebras?

- Poly maps are arrangements, like the above. Set $p := p_1 \otimes \cdots \otimes p_5$.
- A $[p, q]$-coalgebra is a dynamic arr'nt, updating based on what flows.
- Arr'nts are much more general than WDs, e.g. parameters in ANNs.

We can package all this in a monoidal double category called $\mathbb{O}\mathbf{rg}$.

- Its vertical category is **Poly**, e.g. $\text{Ob}(\mathbb{O}\mathbf{rg}) := \text{Ob}(\mathbf{Poly})$.
- For any p, q : **Poly** its category of horizontal morphisms is:

$$\mathbb{O}\mathbf{rg}(p, q) := [p, q]\text{-}\mathbf{coalg}$$

So a horizontal map $p \nrightarrow q$ is a dynamic arrangement of p in q.

- A machine outputting maps $p \to q$ and updating based on what flows.

$\mathbb{O}\mathbf{rg}$ too lives in $\mathbb{C}\mathbf{at}^\sharp$

There is a fully faithful double functor $\mathbb{O}\mathbf{rg} \to \mathbb{C}\mathbf{at}^\sharp$.

- It sends each object $p : \mathbf{Poly}$ to the *cofree comonoid* $\mathfrak{c}_p$ on p.
- Think of this as the cat'y of states and updates for a "p-machine".
- It sends each vertical map $p \xrightarrow{f} q$ to $\mathfrak{c}_p \xrightarrow{\mathfrak{c}_f} \mathfrak{c}_q$.

What does this functor do to a $[p, q]$-coalgebra $S \xrightarrow{\varphi} [p, q](S)$?

- The functor -$\mathbf{coalg}\colon \mathbf{Poly} \to \mathbf{Cat}$ is lax monoidal.
- In particular, we have a map p-$\mathbf{coalg} \times [p, q]$-$\mathbf{coalg} \to q$-$\mathbf{coalg}$.
- So given our $[p, q]$-coalgebra φ, we get a map p-$\mathbf{coalg} \to q$-$\mathbf{coalg}$.
- This turns out to preserve connected limits, hence be a bicomodule.
- Finally, there's an equivalence of categories p-$\mathbf{coalg} \cong \mathfrak{c}_p$-$\mathbf{Set}$.

$$\mathfrak{c}_p \overset{Sy \triangleleft \mathfrak{c}_p}{\triangleright\!\!\longrightarrow\!\!\triangleright} \mathfrak{c}_q$$

So dynamic arrangements (rewiring diagrams) live in $\mathbb{C}\mathbf{at}^\sharp$.

- Thus $\mathbb{C}\mathbf{at}^\sharp$ includes the ANN and prediction market stories.
- And $\mathbb{C}\mathbf{at}^\sharp$ is in some sense just the story of data migration.

Outline

Summary

Poly is an incredibly rich category, and $\mathbb{C}\mathbf{at}^\sharp$ is its comonoids.

- **Poly** is both cartesian closed and monoidal closed; need we say more?
- Comonoids in $(\mathbf{Poly}, y, \triangleleft)$ are exactly categories.
- The comonoid maps and bicomodules make up the equipment $\mathbb{C}\mathbf{at}^\sharp$.

Having unified & ready-made notation, terminology, and techniques is nice.

- That's one thing CT does for math, though it doesn't get everything.
- It is similarly something that $\mathbb{C}\mathbf{at}^\sharp$ does for (A)CT, same caveat.
- Categories, functors, profunctors, cofunctors, pra-functors, dynamic...
- ...arrangements, plus more: nerves of higher categories, rewriting, etc
- It's a setting in which to do formal CT and ACT alike.

Thanks! Comments and questions welcome...