Dynamic Interfaces and Arrangements: An algebraic framework for interacting systems

David I. Spivak

TOPOS INSTITUTE

Category Theory for Consciousness Science
2023 / 04 / 16

Outline

Why am I here?

I think this may be one of the fundamental questions of consciousness.

- In order to flourish, I need to understand my role, how I fit.
- What enabled me and persuaded me to be here?
- This question orients me to the situation and directs my work.
- I'll give an answer below, but first let me reframe "consciousness".

Why am I here?

I think this may be one of the fundamental questions of consciousness.

- In order to flourish, I need to understand my role, how I fit.
- What enabled me and persuaded me to be here?
- This question orients me to the situation and directs my work.
- I'll give an answer below, but first let me reframe "consciousness".

To me, consciousness is *extended*, not isolated within individuals.

- I think of consciousness as that which brings senses into coherence.
- The structure of our brain brings our senses into coherence.
- How much consciousness is in the built environment?
- How much consciousness is in an organization's culture and policies?

Why am I here?

I think this may be one of the fundamental questions of consciousness.

- In order to flourish, I need to understand my role, how I fit.
- What enabled me and persuaded me to be here?
- This question orients me to the situation and directs my work.
- I'll give an answer below, but first let me reframe "consciousness".

To me, consciousness is *extended*, not isolated within individuals.

- I think of consciousness as that which brings senses into coherence.
- The structure of our brain brings our senses into coherence.
- How much consciousness is in the built environment?
- How much consciousness is in an organization's culture and policies?

I'm here because I want a systematic account of collective sense-making.

- How do different sense-makers form into a collective sense-maker?
- Neurons form brains, humans form organizations; we see it all around.
- The senses are not in a heap; they interact and inform each other.
- I want math with which to talk carefully about these ideas.

Accounting

We solve big problems together by coordinating our activity.

- When my efforts and yours conflict, it causes friction and loss.
- When we coordinate, we stop stepping on each others' toes.
- To work collectively, our activities must align.
- We give accounts. We explain our activity in *terms* of the collective.

Accounting

We solve big problems together by coordinating our activity.

- When my efforts and yours conflict, it causes friction and loss.
- When we coordinate, we stop stepping on each others' toes.
- To work collectively, our activities must align.
- We give accounts. We explain our activity in *terms* of the collective.

As the collective matures, its internal accounts become more systematic.

- There is friction every time I misinterpret your account of something.
- Or if your account hides key variables, externalities that I must handle.
- Systematicity increases transparency, communic'n rate, and reliability.
- We become more systematic so that we can regulate each other.

Accounting

We solve big problems together by coordinating our activity.

- When my efforts and yours conflict, it causes friction and loss.
- When we coordinate, we stop stepping on each others' toes.
- To work collectively, our activities must align.
- We give accounts. We explain our activity in *terms* of the collective.

As the collective matures, its internal accounts become more systematic.

- There is friction every time I misinterpret your account of something.
- Or if your account hides key variables, externalities that I must handle.
- Systematicity increases transparency, communic'n rate, and reliability.
- We become more systematic so that we can regulate each other.

Note: regularity is different than predictability.

- A chess game is regular (pawns don't move left), not predictable.
- Regulation: "Hey, you can't move a pawn left"; "Oh, oops!".

Mathematical fields as accounting systems

I think of mathematical fields as crystalized accounting systems.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.

Mathematical fields as accounting systems

I think of mathematical fields as crystalized accounting systems.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.

We want systematic accounting for collective sense-making.

- Math'ns like Newton, Pascal, Frege made new accounting systems.
- Carefully track the phenomena, articulate the structure, systematize.
- So we want to track and articulate the structure of sense-making.

Mathematical fields as accounting systems

I think of mathematical fields as crystalized accounting systems.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.

We want systematic accounting for collective sense-making.

- Math'ns like Newton, Pascal, Frege made new accounting systems.
- Carefully track the phenomena, articulate the structure, systematize.
- So we want to track and articulate the structure of sense-making.

Category theory (CT) is the accounting system for interlocking structures.

- Mathematical definitions are composed of interlocking structures.
- Category theory tracks the layers of structure and their connections.
- This makes analogies—similarities of structure—into formal objects.
- It accounts for the fact that different accounting systems cohere.

Mathematical fields as accounting systems

I think of mathematical fields as crystalized accounting systems.

- Arithmetic accounts for the flow of quantities, as in finance.
- Hilbert spaces account for the states of elementary particles, as in QM.
- Probability distributions account for likelihoods, as in game theory.
- Calculus accounts for relative rates of change.

We want systematic accounting for collective sense-making.

- Math'ns like Newton, Pascal, Frege made new accounting systems.
- Carefully track the phenomena, articulate the structure, systematize.
- So we want to track and articulate the structure of sense-making.

Category theory (CT) is the accounting system for interlocking structures.

- Mathematical definitions are composed of interlocking structures.
- Category theory tracks the layers of structure and their connections.
- This makes analogies—similarities of structure—into formal objects.
- It accounts for the fact that different accounting systems cohere.

Goal: use CT to articulate the structure of collective sense-making.

The morphology of collective sense-making

Collective sense-making—the product of culture—is all around us.

- It's in our science, our technology, our governance, our morality.
- Each of these is the product of our work over millennia.
- Each body is a collective of cells whose individual intelligences...
- ... work harmoniously to create the intelligence at *our level*.

The morphology of collective sense-making

Collective sense-making—the product of culture—is all around us.

- It's in our science, our technology, our governance, our morality.
- Each of these is the product of our work over millennia.
- Each body is a collective of cells whose individual intelligences...
- ... work harmoniously to create the intelligence at *our level*.

I want a language and logic for the *shape* of collective sense-making.

- In particular, I want to be able to talk about this *leveling up*.
- Rather than understanding the lowest level physics...
- ...and relying on "emergence" to get us to human intelligence, ...
- ... I prefer to look for construction principles that are *compositional*.

The morphology of collective sense-making

Collective sense-making—the product of culture—is all around us.

- It's in our science, our technology, our governance, our morality.
- Each of these is the product of our work over millennia.
- Each body is a collective of cells whose individual intelligences...
- ... work harmoniously to create the intelligence at *our level*.

I want a language and logic for the *shape* of collective sense-making.

- In particular, I want to be able to talk about this *leveling up*.
- Rather than understanding the lowest level physics...
- ...and relying on "emergence" to get us to human intelligence, ...
- ... I prefer to look for construction principles that are *compositional*.

Wanted: an algebra by which interacting sense-makers form a sense-maker.

Dynamic organizational systems

Any life-form is a collective, a dynamic organization of smaller parts.

- The organization provides an interaction pattern for the parts.
- The RNA interacts with the nucleus and the ribosome, etc.
- What occurs during these interactions can change the organization.
- As an extreme example, death will allow the system to disintegrate.
- A CEO may see what's occurring and change the company org-chart.

Dynamic organizational systems

Any life-form is a collective, a dynamic organization of smaller parts.

- The organization provides an interaction pattern for the parts.
- The RNA interacts with the nucleus and the ribosome, etc.
- What occurs during these interactions can change the organization.
- As an extreme example, death will allow the system to disintegrate.
- A CEO may see what's occurring and change the company org-chart.

I would like to provide an accounting system for the above idea.

- Open dynamical systems that interact with each other...
- ...according to some pattern: the type of signals/materials that flow.
- The interaction pattern itself can change based on what flows.

Dynamic organizational systems

Any life-form is a collective, a dynamic organization of smaller parts.

- The organization provides an interaction pattern for the parts.
- The RNA interacts with the nucleus and the ribosome, etc.
- What occurs during these interactions can change the organization.
- As an extreme example, death will allow the system to disintegrate.
- A CEO may see what's occurring and change the company org-chart.

I would like to provide an accounting system for the above idea.

- Open dynamical systems that interact with each other...
- ...according to some pattern: the type of signals/materials that flow.
- The interaction pattern itself can change based on what flows.

The CT tool I think we can use is called a *dynamic organizational system.*

- It is based on the theory of polynomial functors.
- Training an ANN (deep learning) is an example of a DOS.
- Other examples: prediction markets, Hebbian learning.
- Can it be extended to collective sense-making? This is open.

Plan

Here is the plan for the rest of the talk.

- Give an account of sense-making and collective intelligence,
- Discuss polynomial functors,
- Introduce dynamic organizational systems,
- Conclude.

Outline

Sense-making: the pun that wasn't

We want to understand collective sense-making.

- But what is *sense*?
- And what does it mean to *make* sense?

Sense-making: the pun that wasn't

We want to understand collective sense-making.

- But what is *sense*?
- And what does it mean to *make* sense?

By *sense*, I don't mean raw perception.

- A spidey sense, a sense of danger or opportunity, a sense of direction.
- These are not mere perception. They're tracking the right variables.
- When we have a good sense of something, we navigate it with ease.

Sense-making: the pun that wasn't

We want to understand collective sense-making.

- But what is *sense*?
- And what does it mean to *make* sense?

By *sense*, I don't mean raw perception.

- A spidey sense, a sense of danger or opportunity, a sense of direction.
- These are not mere perception. They're tracking the right variables.
- When we have a good sense of something, we navigate it with ease.

Consider a snapshot of two math students, both wanting to succeed:

- Student A is faithfully copies down what the teacher says.
- Student B seems to be doing the opposite: ...
- ...clearly frustrated, arguing with the teacher, "but then why XYZ??"
- Suddenly student B says "Oh!! Is it because ABC??"
- B relaxes, having made sense. Later: B does better than A on tests.

Sense-making: the pun that wasn't

We want to understand collective sense-making.

- But what is *sense*?
- And what does it mean to *make* sense?

By *sense*, I don't mean raw perception.

- A spidey sense, a sense of danger or opportunity, a sense of direction.
- These are not mere perception. They're tracking the right variables.
- When we have a good sense of something, we navigate it with ease.

Consider a snapshot of two math students, both wanting to succeed:

- Student A is faithfully copies down what the teacher says.
- Student B seems to be doing the opposite: ...
- ...clearly frustrated, arguing with the teacher, "but then why XYZ??"
- Suddenly student B says "Oh!! Is it because ABC??"
- B relaxes, having made sense. Later: B does better than A on tests.

Making sense of things takes work, but it produces sense!

- The work of trying to make things fit together results in new sense.
- We can solve harder problems if we make better sense of things.

Settling accounts

How are our senses made? Our sense of danger, of sight?

- Could past sense-making activity, installed into deep structures...
- ... account for the senses we have today?
- What could "sense-making" be such that the pun is accurate?

Settling accounts

How are our senses made? Our sense of danger, of sight?

- Could past sense-making activity, installed into deep structures...
- ... account for the senses we have today?
- What could "sense-making" be such that the pun is accurate?

I hypothesize that sense-making has to do with proper accounting.

- When we shake our head and say "that doesn't make sense"...
- ... we're saying it doesn't settle the accounts. Something is left over.
- We jiggle the pieces, try different arrangements until click.
- Something settles. The energy level drops. Delightful!

Settling accounts

How are our senses made? Our sense of danger, of sight?

- Could past sense-making activity, installed into deep structures...
- ... account for the senses we have today?
- What could "sense-making" be such that the pun is accurate?

I hypothesize that sense-making has to do with proper accounting.

- When we shake our head and say "that doesn't make sense"...
- ... we're saying it doesn't settle the accounts. Something is left over.
- We jiggle the pieces, try different arrangements until click.
- Something settles. The energy level drops. Delightful!

Once there's a click, things start to become regular.

- We find an articulation that regularly captures relevant aspects.
- This is exactly the sort of thing we can write down.
- More generally, we can install it into deeper structures.

Consciousness, sense-making, and fit

So far I have made various claims, which I now want to recall.

- I think the question "why am I here?" is fundamental.
- I think of consciousness as that which brings senses into coherence.
- The production of higher-level sense has to do with proper accounting.

Consciousness, sense-making, and fit

So far I have made various claims, which I now want to recall.

- I think the question "why am I here?" is fundamental.
- I think of consciousness as that which brings senses into coherence.
- The production of higher-level sense has to do with proper accounting.

Soon we will be talking about math for dynamic organizations.

- Different dynamic systems interacting within a sort of ecosystem.
- We only have simple examples so far, e.g. ANNs.
- I want help creating a dynamic organizational system of sense-makers.
- If ANNs are optimizing a function, what should sense-makers do?

Consciousness, sense-making, and fit

So far I have made various claims, which I now want to recall.

- I think the question "why am I here?" is fundamental.
- I think of consciousness as that which brings senses into coherence.
- The production of higher-level sense has to do with proper accounting.

Soon we will be talking about math for dynamic organizations.

- Different dynamic systems interacting within a sort of ecosystem.
- We only have simple examples so far, e.g. ANNs.
- I want help creating a dynamic organizational system of sense-makers.
- If ANNs are optimizing a function, what should sense-makers do?

If sense-makers want to cohere, they should understand their own fit.

- Etymologically, fitness means "the quality of fitting".
- When the sense-maker understands math, they see how it all fits.
- "Why am I here?" asks "how do I fit"? "What is my role?"
- If each member of a collective has a good sense of their own fit,...
- ...it creates coherence, establishing higher-order (collective) sense.

How the math fits in

I was trained as a mathematician, not as a philosopher.

- My role here is not to philosophize all hour, but to present some math.
- The math is intended as an accounting system for something relevant.
- Namely, it accounts for coherent interaction of dynamical systems.

How the math fits in

I was trained as a mathematician, not as a philosopher.

- My role here is not to philosophize all hour, but to present some math.
- The math is intended as an accounting system for something relevant.
- Namely, it accounts for coherent interaction of dynamical systems.

I'll next introduce the main tool: polynomial functors.

- Polynomial functors—despite the boring name—are *stunning*.
- They're the most highly structured and...
- ...unreasonably effective abstraction I've ever seen.

How the math fits in

I was trained as a mathematician, not as a philosopher.

- My role here is not to philosophize all hour, but to present some math.
- The math is intended as an accounting system for something relevant.
- Namely, it accounts for coherent interaction of dynamical systems.

I'll next introduce the main tool: polynomial functors.

- Polynomial functors—despite the boring name—are *stunning*.
- They're the most highly structured and...
- ...unreasonably effective abstraction I've ever seen.

Polynomial functors form the basis for dynamic organizational systems.

- I'm going to explain polynomial functors at many levels simultaneously.
- You may not understand certain ideas/words; just let them go.
- I won't leave you long without something you can make sense of.

Outline

Unreasonable effectiveness

Wigner lauded math as *unreasonably effective* in the natural sciences.

- Many of his assertions also affirm the effectiveness of CT in math.
- He mentions the miracle that is our ability to make sense of the world.

Unreasonable effectiveness

Wigner lauded math as *unreasonably effective* in the natural sciences.

- Many of his assertions also affirm the effectiveness of CT in math.
- He mentions the miracle that is our ability to make sense of the world.

Probably the real miracle here is *abstraction*, a bi-directional thing:

- We can take a concrete situation and boil it down to an abstract one.
- This first part can be imagined as a function $b\colon C \to A$.
- Then we can take conclusions about the boiled down $b(c) : A$ and...
- ... transport them back to the concrete situation c we started with.

Unreasonable effectiveness

Wigner lauded math as *unreasonably effective* in the natural sciences.

- Many of his assertions also affirm the effectiveness of CT in math.
- He mentions the miracle that is our ability to make sense of the world.

Probably the real miracle here is *abstraction*, a bi-directional thing:

- We can take a concrete situation and boil it down to an abstract one.
- This first part can be imagined as a function $b\colon C \to A$.
- Then we can take conclusions about the boiled down $b(c) : A$ and...
- ... transport them back to the concrete situation c we started with.

I think **Poly** is similarly unreasonably effective for computer science.

- The category **Poly** is strange but still pretty easy to think about.
- In some sense it's all about plumbing abstractions.
- It's got tons of structure: limits, colimits, three orthogonal factorization systems, infinitely-many monoidal closed structures, various coclosures, its comonoids are categories, its monoids generalize operads, etc.
- But it also has tons of applications in CS: Moore machines and Mealy machines, databases and data migration, algebraic datatypes, bi-directional transformations, dependent type theory, effects handling, cellular automata, rewriting workflows, deep learning.

Unreasonable effectiveness

Wigner lauded math as *unreasonably effective* in the natural sciences.

- Many of his assertions also affirm the effectiveness of CT in math.
- He mentions the miracle that is our ability to make sense of the world.

Probably the real miracle here is *abstraction*, a bi-directional thing:

- We can take a concrete situation and boil it down to an abstract one.
- This first part can be imagined as a function $b\colon C \to A$.
- Then we can take conclusions about the boiled down $b(c) : A$ and...
- ... transport them back to the concrete situation c we started with.

I think **Poly** is similarly unreasonably effective for computer science.

- The category **Poly** is strange but still pretty easy to think about.
- In some sense it's all about plumbing abstractions.
- It's got tons of structure: limits, colimits, three orthogonal factorization systems, infinitely-many monoidal closed structures, various coclosures, its comonoids are categories, its monoids generalize operads, etc.
- But it also has tons of applications in CS: Moore machines and Mealy machines, databases and data migration, algebraic datatypes, bi-directional transformations, dependent type theory, effects handling, cellular automata, rewriting workflows, deep learning.

So what are polynomials?

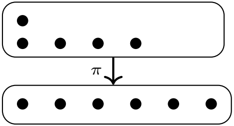
Definition and intuition

A *polynomial p* is essentially a data structure. Here are three viewpoints:

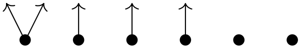
Algebraic

$$y^2 + 3y + 2$$

Bundle



Corolla forest



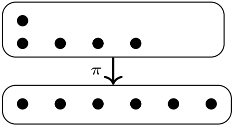
Definition and intuition

A *polynomial* p is essentially a data structure. Here are three viewpoints:

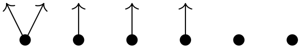
Algebraic

$$y^2 + 3y + 2$$

Bundle



Corolla forest



Cat. description: **Poly** = "sums of representables functors **Set** → **Set**".

- For any set S, let $y^S := \mathbf{Set}(S, -)$, the functor *represented* by S.
- Def: a polynomial is a sum $p = \sum_{i:I} y^{P_i}$ of representable functors.
- Def: a morphism of polynomials is a natural transformation.
- Note that $I = p(1)$; this is a convenient fact. Write $p[i]$ for P_i.

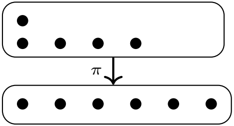
Definition and intuition

A *polynomial* p is essentially a data structure. Here are three viewpoints:

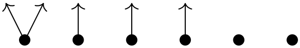
Algebraic

$y^2 + 3y + 2$

Bundle



Corolla forest



Cat. description: **Poly** = "sums of representables functors **Set** → **Set**".

- For any set S, let $y^S := \mathbf{Set}(S, -)$, the functor *represented* by S.
- Def: a polynomial is a sum $p = \sum_{i:I} y^{P_i}$ of representable functors.
- Def: a morphism of polynomials is a natural transformation.
- Note that $I = p(1)$; this is a convenient fact. Write $p[i]$ for P_i.

Other ways to see a polynomial $p = \sum_{i:I} y^{p[i]}$ as an interface:

- A set I of *types*; each type $i : I$ has a set $p[i]$ of *terms*.
- A set I of *problems*; each problem $i : I$ has a set $p[i]$ of *solutions*.
- A set I of *body positions*; each pos'n $i : I$ has a set $p[i]$ of *sensations*.

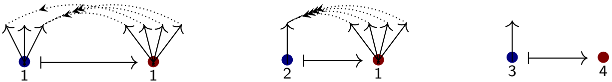
Combinatorics of polynomial morphisms

Let $p := y^3 + 2y$ and $q := y^4 + y^2 + 2$

Combinatorics of polynomial morphisms

Let $p \coloneqq y^3 + 2y$ and $q \coloneqq y^4 + y^2 + 2$

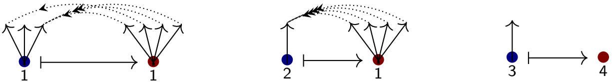
A morphism $p \xrightarrow{\varphi} q$ delegates each p-position to a q-position, passing back directions:



Combinatorics of polynomial morphisms

Let $p := y^3 + 2y$ and $q := y^4 + y^2 + 2$

A morphism $p \xrightarrow{\varphi} q$ delegates each p-position to a q-position, passing back directions:



- It's like we said about abstraction. $\varphi\colon p \to q$ means: ...
- ...φ abstracts each problem in p to one in q, and...
- ...φ then implements each q-solution as a p-solution.

Operations: $+, \times, \otimes, \triangleleft, [-, -]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p + q$: problem is $i : p(1)$ or $j : q(1)$; solve it.

Operations: $+, \times, \otimes, \triangleleft, [-,-]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p + q$: problem is $i : p(1)$ or $j : q(1)$; solve it.
- Product $p \times q$: problem is pair $(i, j) : p(1) \times q(1)$; solve either.

Operations: $+, \times, \otimes, \triangleleft, [-,-]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p + q$: problem is $i : p(1)$ or $j : q(1)$; solve it.
- Product $p \times q$: problem is pair $(i, j) : p(1) \times q(1)$; solve either.
- Dirichlet product $p \otimes q$: prob'm is pair $(i, j) : p(1) \times q(1)$; solve both.

Operations: $+, \times, \otimes, \triangleleft, [-,-]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p + q$: problem is $i : p(1)$ or $j : q(1)$; solve it.
- Product $p \times q$: problem is pair $(i, j) : p(1) \times q(1)$; solve either.
- Dirichlet product $p \otimes q$: prob'm is pair $(i, j) : p(1) \times q(1)$; solve both.
- Substitution product $p \triangleleft q$: prob'm is choice of $i : p(1)$ and...
- ...for every solution a problem $j : q(1)$; solve first then second.

Operations: $+, \times, \otimes, \triangleleft, [-,-]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p + q$: problem is $i : p(1)$ or $j : q(1)$; solve it.
- Product $p \times q$: problem is pair $(i, j) : p(1) \times q(1)$; solve either.
- Dirichlet product $p \otimes q$: prob'm is pair $(i, j) : p(1) \times q(1)$; solve both.
- Substitution product $p \triangleleft q$: prob'm is choice of $i : p(1)$ and...
- ...for every solution a problem $j : q(1)$; solve first then second.
- Internal hom $[p, q]$: problem is polynomial map $\varphi \colon p \to q$;...
- ...soln: problem $i : p(1)$ and solution to its image $\varphi_1(i) : q(1)$.

Operations: $+, \times, \otimes, \triangleleft, [-,-]$

Given two interfaces p, q, there are many ways to get another interface.

- For each we'll say the problems and solutions for resulting interface.
- Sum $p+q$: problem is $i : p(1)$ or $j : q(1)$; solve it.
- Product $p \times q$: problem is pair $(i,j) : p(1) \times q(1)$; solve either.
- Dirichlet product $p \otimes q$: prob'm is pair $(i,j) : p(1) \times q(1)$; solve both.
- Substitution product $p \triangleleft q$: prob'm is choice of $i : p(1)$ and...
- ...for every solution a problem $j : q(1)$; solve first then second.
- Internal hom $[p,q]$: problem is polynomial map $\varphi\colon p \to q$;...
- ...soln: problem $i : p(1)$ and solution to its image $\varphi_1(i) : q(1)$.

Letting $p \coloneqq \sum_{i:p(1)} y^{p_i}$ and $q \coloneqq \sum_{j:q(1)} y^{q_j}$

$$p \times q = \sum_{(i,j)} y^{p[i]+q[j]} \qquad p \otimes q = \sum_{(i,j)} y^{p[i]\times q[j]}$$

$$p \triangleleft q = \sum_{i:p(1)} \sum_{j\colon p[i] \to q(1)} y^{\sum_{x:p[i]} q[jx]} \qquad [p,q] = \sum_{\varphi\colon p \to q} y^{\sum_{i:p(1)} q[\varphi_1 i]}$$

Open dynamical systems: Moore machines

We will be interested in open dynamical systems.

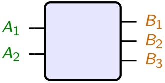
- An open dynamical system has an interface, which we draw as a box.

- A, B are sets, the set of things that can flow on the wire.
- The input ports are drawn on the left and output on the right.

Open dynamical systems: Moore machines

We will be interested in open dynamical systems.

- An open dynamical system has an interface, which we draw as a box.



- A, B are sets, the set of things that can flow on the wire.
- The input ports are drawn on the left and output on the right.

An (A, B)-dynamical system has *internal states*, which govern its behavior.

- That is, it includes a set S : **Set** and two functions:
- a function $\varphi^{\text{rdt}}\colon S \to B$ called *readout* and
- a function $\varphi^{\text{upd}}\colon S \times A \to S$ called *update*.

Open dynamical systems: Moore machines

We will be interested in open dynamical systems.

- An open dynamical system has an interface, which we draw as a box.

- A, B are sets, the set of things that can flow on the wire.
- The input ports are drawn on the left and output on the right.

An (A, B)-dynamical system has *internal states*, which govern its behavior.

- That is, it includes a set S : **Set** and two functions:
- a function $\varphi^{\mathrm{rdt}} \colon S \to B$ called *readout* and
- a function $\varphi^{\mathrm{upd}} \colon S \times A \to S$ called *update*.

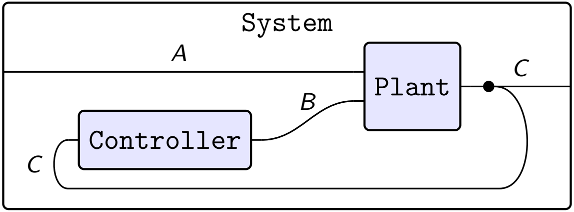
All this is called a Moore machine and is nicely represented in **Poly**.

- The interface is represented by the polynomial By^A or $B_1 B_2 B_3 y^{A_1 A_2}$.
- The readout and update are defined by a single polynomial map

$$\varphi \colon Sy^S \to By^A$$

Wiring diagrams

Here's a picture of a wiring diagram:

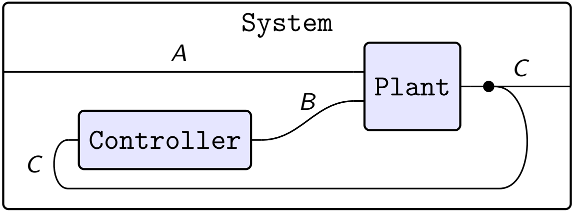


It includes three interfaces: Controller, Plant, and System.

$$\texttt{Controller} = By^{C} \qquad \texttt{Plant} = Cy^{AB} \qquad \texttt{System} = Cy^{A}$$

Wiring diagrams

Here's a picture of a wiring diagram:



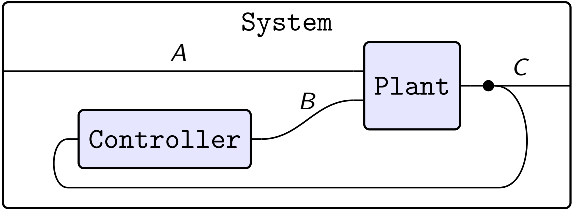
It includes three interfaces: Controller, Plant, and System.

$$\texttt{Controller} = By^C \qquad \texttt{Plant} = Cy^{AB} \qquad \texttt{System} = Cy^A$$

The wiring diagram represents a map $\texttt{Controller} \otimes \texttt{Plant} \to \texttt{System}$.

$$By^C \otimes Cy^{AB} \longrightarrow Cy^A$$

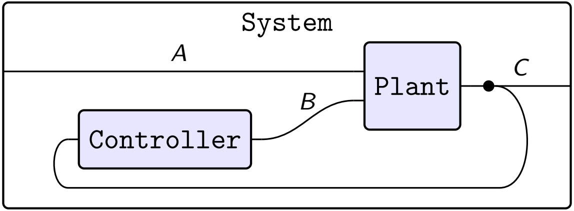
Moore machines and wiring diagrams as lenses



To summarize what we've said so far:

- A wiring diagram (WD) is a map, e.g. $By^C \otimes Cy^{AB} \longrightarrow Cy^A$.
- Each Moore machine is a map, e.g. $Sy^S \to By^C$ and $Ty^T \to Cy^{AB}$.

Moore machines and wiring diagrams as lenses



To summarize what we've said so far:

- A wiring diagram (WD) is a map, e.g. $By^C \otimes Cy^{AB} \longrightarrow Cy^A$.
- Each Moore machine is a map, e.g. $Sy^S \to By^C$ and $Ty^T \to Cy^{AB}$.

We can tensor the Moore machines and compose to obtain $STy^{ST} \to Cy^A$.

- So a wiring diagram is a formula for combining Moore machines.
- The whole story is polynomials, through and through.
- So far, all the polynomials we've been using are monomials Ay^B.
- For "mode dependence" where interfaces can change, use gen'l polys.

Moore machines, Mealy machines, and coalgebras

There's a little more to say about open dynamical systems.

- We just said that an (A, B)-Moore machine is a map $Sy^S \to By^A$.
- This is equivalent to a more common cat'ical approach: coalgebras.
- An (A, B)-Moore machine is equivalently a function $S \to By^A \triangleleft S$.

Moore machines, Mealy machines, and coalgebras

There's a little more to say about open dynamical systems.

- We just said that an (A, B)-Moore machine is a map $Sy^S \to By^A$.
- This is equivalent to a more common cat'ical approach: coalgebras.
- An (A, B)-Moore machine is equivalently a function $S \to By^A \triangleleft S$.

There's another whole type of dynamical system: Mealy machines.

- The two are actually inter-convertible, but they have different forms.

Moore: $S \to B, \quad S \times A \to S$ | Mealy: $S \times A \to S \times B$

$$Sy^S \to By^A \qquad\qquad Sy^S \to (By)^A$$

- To get from input to output takes one step in Moore, instant in Mealy.
- An (A, B)-Moore machine is a special (A, B)-Mealy machine.
- An (A, B)-Mealy machine is exactly an (A, B^A)-Moore machine.

Outline

Categories where the morphisms are changing

Imagine something like **Set**, except that morphisms are dynamic.

- For sets A, B, a morphism $f\colon A \to B$ is a machine with states S.
- In its current state $s : S$, it outputs an actual function $f_s\colon A \to B$.
- Given an input $a : A$, it not only tells you $f_s(a)$ but *updates its state*.
- I want to call refer to a morphism f as a *dynamic function*.

Categories where the morphisms are changing

Imagine something like **Set**, except that morphisms are dynamic.

- For sets A, B, a morphism $f\colon A \to B$ is a machine with states S.
- In its current state $s : S$, it outputs an actual function $f_s\colon A \to B$.
- Given an input $a : A$, it not only tells you $f_s(a)$ but *updates its state*.
- I want to call refer to a morphism f as a *dynamic function*.

We've actually already seen these: they're the (A, B)-Mealy machines.

- That is, they are the functions $f\colon S \times A \to S \times B$.
- This fits into a more general **Poly** story, namely using internal homs.
- I'll spare you the details, but here's the basic idea:
    - For any p, q : **Poly**, a $[p, q]$-coalgebra is a dyn'l system that...
    - ...outputs interaction patterns $p \to q$ (e.g. any wiring diagram)...
    - ...and updates internal state based on what flows along the wires.
    - Again, in the case $p = A\mathcal{y}$ and $q = B\mathcal{y}$, you get Mealy machines.

Categories where the morphisms are changing

Imagine something like **Set**, except that morphisms are dynamic.

- For sets A, B, a morphism $f\colon A \to B$ is a machine with states S.
- In its current state $s : S$, it outputs an actual function $f_s\colon A \to B$.
- Given an input $a : A$, it not only tells you $f_s(a)$ but *updates its state*.
- I want to call refer to a morphism f as a *dynamic function*.

We've actually already seen these: they're the (A, B)-Mealy machines.

- That is, they are the functions $f\colon S \times A \to S \times B$.
- This fits into a more general **Poly** story, namely using internal homs.
- I'll spare you the details, but here's the basic idea:
  - For any p, q : **Poly**, a $[p, q]$-coalgebra is a dyn'l system that...
  - ...outputs interaction patterns $p \to q$ (e.g. any wiring diagram)...
  - ...and updates internal state based on what flows along the wires.
  - Again, in the case $p = A\mathcal{y}$ and $q = B\mathcal{y}$, you get Mealy machines.

Two more technical slides.

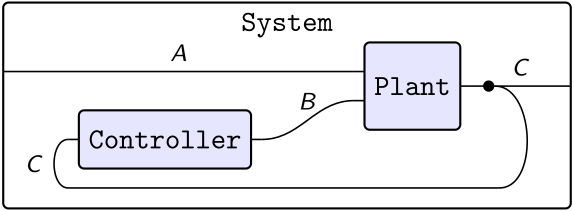
Definition of $\mathbb{O}\mathbf{rg}$

We can now define the bicategory $\mathbb{O}\mathbf{rg}$.

- $\mathrm{Ob}(\mathbb{O}\mathbf{rg}) \coloneqq \mathrm{Ob}(\mathbf{Poly})$, objects are polynomials.
- $\mathbb{O}\mathbf{rg}(p, q) \coloneqq [p, q]\text{-}\mathbf{coalg}$.

Example: suppose $p = By^C \otimes Cy^{AB}$ and $q = Cy^A$.

- Then for any state $s : S$ of a $[p, q]$-coalgebra (S, f), we have first...
- ...a map $p \to q$. For example, we may have this one:



- That is, we're outputting interaction patterns. We have second,...

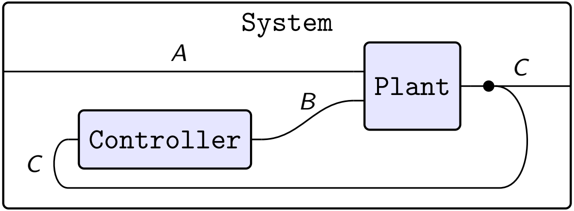
Definition of $\mathbb{O}\mathbf{rg}$

We can now define the bicategory $\mathbb{O}\mathbf{rg}$.

- $\text{Ob}(\mathbb{O}\mathbf{rg}) \coloneqq \text{Ob}(\mathbf{Poly})$, objects are polynomials.
- $\mathbb{O}\mathbf{rg}(p, q) \coloneqq [p, q]$-**coalg**.

Example: suppose $p = By^C \otimes Cy^{AB}$ and $q = Cy^A$.

- Then for any state $s : S$ of a $[p, q]$-coalgebra (S, f), we have first...
- ...a map $p \to q$. For example, we may have this one:



- That is, we're outputting interaction patterns. We have second,...
- ...a state update function whose input is "what flows on the wires".
- So (S, f) outputs interaction patterns and listens to what flows.

ANNs in terms of $\mathbb{O}$rg

We can now describe artificial neural networks in this language.

- Let $t \coloneqq \sum_{x \in \mathbb{R}} y^{T_x^* \mathbb{R}} \cong \mathbb{R} y^{\mathbb{R}}$.
- So "positions of t" = points in $\mathbb{R}$ and "directions" = gradients.
- Note that $t \otimes t \cong \sum_{x \in \mathbb{R}^2} y^{T_x^* \mathbb{R}^2} \cong \mathbb{R}^2 y^{\mathbb{R}^2}$ and similarly for any $t^{\otimes n}$.

ANNs in terms of $\mathbb{O}$rg

We can now describe artificial neural networks in this language.

- Let $t \coloneqq \sum_{x \in \mathbb{R}} y^{T_x^* \mathbb{R}} \cong \mathbb{R} y^{\mathbb{R}}$.
- So "positions of t" = points in $\mathbb{R}$ and "directions" = gradients.
- Note that $t \otimes t \cong \sum_{x \in \mathbb{R}^2} y^{T_x^* \mathbb{R}^2} \cong \mathbb{R}^2 y^{\mathbb{R}^2}$ and similarly for any $t^{\otimes n}$.

A $[t^{\otimes m}, t^{\otimes n}]$-coalgebra consists of:

- A set S of states / parameters / weights&biases, and for each $s : S$...
- ... a function $f_s \colon \mathbb{R}^m \to \mathbb{R}^n$ and ...
- ... a function $(x : \mathbb{R}^m) \times (y' : T^*_{f_s(x)} \mathbb{R}^n) \to S \times T^*_s \mathbb{R}^m$.

ANNs in terms of $\mathbb{O}$rg

We can now describe artificial neural networks in this language.

- Let $t := \sum_{x \in \mathbb{R}} y^{T_x^* \mathbb{R}} \cong \mathbb{R} y^{\mathbb{R}}$.
- So "positions of t" = points in $\mathbb{R}$ and "directions" = gradients.
- Note that $t \otimes t \cong \sum_{x \in \mathbb{R}^2} y^{T_x^* \mathbb{R}^2} \cong \mathbb{R}^2 y^{\mathbb{R}^2}$ and similarly for any $t^{\otimes n}$.

A $[t^{\otimes m}, t^{\otimes n}]$-coalgebra consists of:

- A set S of states / parameters / weights&biases, and for each $s : S$...
- ... a function $f_s \colon \mathbb{R}^m \to \mathbb{R}^n$ and ...
- ... a function $(x : \mathbb{R}^m) \times (y' : T^*_{f_s(x)} \mathbb{R}^n) \to S \times T^*_s \mathbb{R}^m$.

This latter thing might be called "update and backprop".

- It takes an input $x : \mathbb{R}^m$ and a gradient $y' : T^*_{f(s)} \mathbb{R}^n$ and returns...
- ...a new/updated state $s' : S$ and a backprop'd gradient $x' : T^*_s \mathbb{R}^m$.

ANNs in terms of $\mathbb{O}$rg

We can now describe artificial neural networks in this language.

- Let $t := \sum_{x \in \mathbb{R}} y^{T_x^* \mathbb{R}} \cong \mathbb{R} y^{\mathbb{R}}$.
- So "positions of t" = points in $\mathbb{R}$ and "directions" = gradients.
- Note that $t \otimes t \cong \sum_{x \in \mathbb{R}^2} y^{T_x^* \mathbb{R}^2} \cong \mathbb{R}^2 y^{\mathbb{R}^2}$ and similarly for any $t^{\otimes n}$.

A $[t^{\otimes m}, t^{\otimes n}]$-coalgebra consists of:

- A set S of states / parameters / weights&biases, and for each $s : S$...
- ... a function $f_s \colon \mathbb{R}^m \to \mathbb{R}^n$ and ...
- ... a function $(x : \mathbb{R}^m) \times (y' : T^*_{f_s(x)} \mathbb{R}^n) \to S \times T^*_s \mathbb{R}^m$.

This latter thing might be called "update and backprop".

- It takes an input $x : \mathbb{R}^m$ and a gradient $y' : T^*_{f(s)} \mathbb{R}^n$ and returns...
- ...a new/updated state $s' : S$ and a backprop'd gradient $x' : T^*_s \mathbb{R}^m$.

There are many such $[t^{\otimes m}, t^{\otimes n}]$-coalgebras.

- One has carrier $S := \{P : \mathbb{N}, f : \mathbb{R}^P \times \mathbb{R}^m \to \mathbb{R}^n \text{ diff'ntiable}, p : \mathbb{R}^P\}$.
- The state (P, f, p) updated by training pair $(x : \mathbb{R}^m, y' : T^*_{f(p,x)} \mathbb{R}^n)$
- ... is (P, f, p') where $p' := p + \pi_P \left(Df^\top_{(p,x)} \cdot y' \right)$.

Model of prediction markets

Let's consider a simple version of a prediction market. Suppose:

- There is a fixed finite set X of outcomes.
- Each participant can output a prediction $P : \Delta_+(X)$ where

$$\Delta_+(X) := \left\{ P : X \to (0,1] \;\middle|\; 1 = \sum_{x \in X} P(x) \right\}$$

- Each participant then receives the result, an element $x : X$.

Model of prediction markets

Let's consider a simple version of a prediction market. Suppose:

- There is a fixed finite set X of outcomes.
- Each participant can output a prediction $P : \Delta_+(X)$ where

$$\Delta_+(X) := \left\{ P : X \to (0,1] \;\middle|\; 1 = \sum_{x \in X} P(x) \right\}$$

- Each participant then receives the result, an element $x : X$.

It's compositional if we assign predictors a relative "trust" / "wealth".

- Let n be a finite set of predictors. A relative trust is $t : \Delta(n)$.
- Given $n : \mathbb{N}$, $t : \Delta(n)$, and predictors $P_1, \dots, P_n : \Delta_+(X)$, ...
- ...we get a new predictor $t \cdot P = t(1) * P_1 + \cdots + t(n) * P_n$.
- I.e., we multiply each prediction by how much we trust its predictor.

Prediction markets in terms of $\mathbb{O}$rg

Fix X : **Fin**. We use the polynomial $p \coloneqq \Delta_+(X)y^X$ to model a predictor.

- It outputs a prediction $P : \Delta_+(X)$ and inputs an actual outcome $x : X$.
- Then $p^{\otimes n}$ outputs n predictions and receives n outcomes.
- Consider the polynomial $[p^{\otimes n}, p]$. A position includes:...
- ...a function $\Delta_+(X)^n \to \Delta_+(X)$, and a function $X \to X^n$. ...
- It's a way to combine n predictions into one and distribute outcomes.
- A direction of $[p^{\otimes n}, p]$ consists of: n-many pred'ns and one outcome.

Prediction markets in terms of $\mathbb{O}$rg

Fix $X : \mathbf{Fin}$. We use the polynomial $p \coloneqq \Delta_+(X)y^X$ to model a predictor.

- It outputs a prediction $P : \Delta_+(X)$ and inputs an actual outcome $x : X$.
- Then $p^{\otimes n}$ outputs n predictions and receives n outcomes.
- Consider the polynomial $[p^{\otimes n}, p]$. A position includes:...
- ...a function $\Delta_+(X)^n \to \Delta_+(X)$, and a function $X \to X^n$. ...
- It's a way to combine n predictions into one and distribute outcomes.
- A direction of $[p^{\otimes n}, p]$ consists of: n-many pred'ns and one outcome.

The category of maps $p^{\otimes n} \to p$ in $\mathbb{O}$**rg** is $[p^{\otimes n}, p]$-**coalg**.

- Such a coalgebra consists of a set T_n and for each $t : T_n$,...
- ...a function $\Delta_+(X)^n \to \Delta_+(X)$, a function $X \to X^n$, and...
- ...given n predictions $P_1, \ldots, P_n$ and an outcome x, a new state.

Prediction markets in terms of $\mathbb{O}$rg

Fix X : **Fin**. We use the polynomial $p := \Delta_+(X)y^X$ to model a predictor.

- It outputs a prediction $P : \Delta_+(X)$ and inputs an actual outcome $x : X$.
- Then $p^{\otimes n}$ outputs n predictions and receives n outcomes.
- Consider the polynomial $[p^{\otimes n}, p]$. A position includes:...
- ...a function $\Delta_+(X)^n \to \Delta_+(X)$, and a function $X \to X^n$. ...
- It's a way to combine n predictions into one and distribute outcomes.
- A direction of $[p^{\otimes n}, p]$ consists of: n-many pred'ns and one outcome.

The category of maps $p^{\otimes n} \to p$ in $\mathbb{O}$**rg** is $[p^{\otimes n}, p]$-**coalg**.

- Such a coalgebra consists of a set T_n and for each $t : T_n$,...
- ...a function $\Delta_+(X)^n \to \Delta_+(X)$, a function $X \to X^n$, and...
- ...given n predictions $P_1, \ldots, P_n$ and an outcome x, a new state.

There are many such coalgebras. The one for us is:

- Take $T_n := \Delta_n$, the set of "relative trust levels" for n players.
- Given $t : T_n$, use $t \cdot - : \Delta_+(X)^n \to \Delta_+(X)$ and $x \mapsto (x, x, \ldots, x)$.
- Given pred'ns $(P_i)_{i:n}$ and outcome x, use Bayesian upd. to get new t'.

Dynamic organizational systems

So what are dynamic organizational systems?[1]

- We've shown two examples: ANNs and prediction markets.
- Technically, these are monoidal caty's or operads enriched in $\mathbb{O}$**rg**.
- A single procedure (e.g. gradient descent, Bayesian update)...
- ...which can be performed locally (per neuron, per predictor)...
- ...such that composites of this procedure again perform the procedure.

[1] This is joint work with Brandon Shapiro (arXiv:2205.03906).

Dynamic organizational systems

So what are dynamic organizational systems?[1]

- We've shown two examples: ANNs and prediction markets.
- Technically, these are monoidal caty's or operads enriched in $\mathbb{O}\mathbf{rg}$.
- A single procedure (e.g. gradient descent, Bayesian update)...
- ...which can be performed locally (per neuron, per predictor)...
- ...such that composites of this procedure again perform the procedure.

And why are they relevant to consciousness?

- I'd like someone to define a dynamic sense-making system.
- It organizes itself (like an ANN or pred'n market) through experience.
- Q: what single procedure, performed locally (per sense-maker)...
- ...would make a composite of sense-makers again be a sense-maker?
- I imagine each trying to account for the environment and its own fit.
- I imagine the accounting language naturally becoming more systematic

[1] This is joint work with Brandon Shapiro (arXiv:2205.03906).

Outline

Summary

We ask how sense is *made*.

- Sense of danger, direction, humor: how to track the "right" variables?
- We make sense by settling accounts: everything fits into place.
- We work to install sense-making systems into the deepest structures.

Summary

We ask how sense is *made*.

- Sense of danger, direction, humor: how to track the "right" variables?
- We make sense by settling accounts: everything fits into place.
- We work to install sense-making systems into the deepest structures.

Mathematics is highly systematic (crystalized) accounting.

- We use it to give very structured, repeatable, regulatable accounts.
- The math guides our questioning and makes results communicable.
- Category theory is the accounting system for interlocking structures.
- **Poly** is a stunningly structured category, unreasonably effective in CS.

Summary

We ask how sense is *made*.

- Sense of danger, direction, humor: how to track the "right" variables?
- We make sense by settling accounts: everything fits into place.
- We work to install sense-making systems into the deepest structures.

Mathematics is highly systematic (crystalized) accounting.

- We use it to give very structured, repeatable, regulatable accounts.
- The math guides our questioning and makes results communicable.
- Category theory is the accounting system for interlocking structures.
- **Poly** is a stunningly structured category, unreasonably effective in CS.

Dynamic organizational systems are ways for local entities to self-organize.

- ANNs and pred'n markets self-organize based on training / experience.
- Open question: define a DOS for sense-making?
- The math guides questioning about how we *make* sense.

Thanks! Comments and questions welcome...