Learners' languages

David I. Spivak

TOPOS INSTITUTE

Topos Internal Seminar
2021 September 21

Outline

Goal of today's talk

I want to tell you what I've been doing and where I'm stuck.

- Thanks Toby, for saying that that's what you wanted to hear about.

I feel like I've developed the machinery I want to use.

- As you know, I love Poly: it's expressive and well-behaved.
- Inside Poly is a categorical operad I'm calling $\mathcal{O}\mathcal{r}\mathcal{g}$.
- It packages my usual "interacting machines" thing inside of Poly.

Goal of today's talk

I want to tell you what I've been doing and where I'm stuck.

- Thanks Toby, for saying that that's what you wanted to hear about.

I feel like I've developed the machinery I want to use.

- As you know, I love Poly: it's expressive and well-behaved.
- Inside Poly is a categorical operad I'm calling $\mathcal{O}\mathfrak{r}g$.
- It packages my usual "interacting machines" thing inside of Poly.

But now I have to actually use it.

- I want to talk about what matters most to me:
- What actually makes things work?
- What is coordination, cooperation, health, effectiveness?

Plan of the talk

- Background on Poly.
  - The basics.
  - The $(y, \otimes)$ monoidal structure and its closure $[-, -]$.
  - Coalgebras: e.g. dynamical systems and wiring diagrams.
  - Type theory and logic in the topos of *p*-coalgebras.
- Introduce the categorical operad $\mathcal{O}\mathcal{r}\mathcal{g}$ of *organizations*.
  - Recall the setup in “backprop as functor”.
  - Define $\mathcal{O}\mathcal{r}\mathcal{g}$ and give intuition.
- Explain where I’m stuck.

Outline

Definition and terminology of Poly

Poly is the category of sums of representables Set $\to$ Set.

- For any $A \in$ Set, write $y^A \in$ Poly to mean $(X \mapsto X^A)$: Set $\to$ Set.
- A polynomial p is a coproduct of representables, $p = \sum_{i \in I} y^{A_i}$.
  - Call each $i \in I$ a *position in p*.
  - Note: $p(1) \cong I$.
  - Call each $a \in A_i$ a *direction in p* (at position i).
- Let's write $p[i]$ instead of A_i, to obtain this notation:

$$p = \sum_{i \in p(1)} y^{p[i]}$$

Definition and terminology of Poly

Poly is the category of sums of representables Set $\to$ Set.

- For any $A \in$ Set, write $y^A \in$ Poly to mean $(X \mapsto X^A)$: Set $\to$ Set.
- A polynomial p is a coproduct of representables, $p = \sum_{i \in I} y^{A_i}$.
  - Call each $i \in I$ a *position in p*.
  - Note: $p(1) \cong I$.
  - Call each $a \in A_i$ a *direction in p* (at position i).
- Let's write $p[i]$ instead of A_i, to obtain this notation:

$$p = \sum_{i \in p(1)} y^{p[i]}$$

Morphisms $\varphi\colon p \to q$ in Poly are just natural transformations.

- Yoneda: $\mathrm{Poly}(y^A, y^B) = \mathrm{Set}(B, A)$.
- Derive "lens-like" description from universal property of coproducts:

$$\varphi \in \mathrm{Poly}\Big(\sum_{i \in p(1)} y^{p[i]}, \sum_{j \in q(1)} y^{q[j]}\Big) \cong \prod_{i \in p(1)} \sum_{j \in q(1)} \mathrm{Set}(q[j], p[i])$$

The monoidal structure $(y, \otimes)$

There is a monoidal structure $(y, \otimes)$ on Poly.

- $y \in \mathsf{Poly}$ denotes the identity functor $\mathrm{id}\colon \mathsf{Set} \to \mathsf{Set}$.
    - The polynomial y has one position, and one direction.
    - Quick aside on maps into y:
        - A map $\gamma\colon p \to y$ is a "global section" of p.
        - That is, it's a choice of direction at each position.

The monoidal structure $(y, \otimes)$

There is a monoidal structure $(y, \otimes)$ on Poly.

- $y \in$ Poly denotes the identity functor id: Set $\to$ Set.
  - The polynomial y has one position, and one direction.
  - Quick aside on maps into y:
    - A map $\gamma\colon p \to y$ is a "global section" of p.
    - That is, it's a choice of direction at each position.
  - Back to the main point: y is the unit of a monoidal structure.
- Given polynomials p, q, we can multiply both positions and directions.

$$p \otimes q := \sum_{i \in p(1)} \sum_{j \in q(1)} y^{p[i] \times q[j]}$$

- Examples:
  - $A \otimes B = AB$
  - $Ay \otimes By = ABy$
  - $y^A \otimes y^B = y^{AB}$
  - $p \otimes 1 = p(1)$

The $\otimes$-closure, i.e. internal hom $[-,-]$

For any two polynomials $p, q \in \mathsf{Poly}$, there is $[p, q] \in \mathsf{Poly}$ with

$$\mathsf{Poly}(p' \otimes p, q) \cong \mathsf{Poly}(p', [p, q])$$

for any $p' \in \mathsf{Poly}$. It can be given by the following formula:

$$[p, q] := \sum_{\varphi \colon p \to q} y^{\sum_{i \in p(1)} q[\varphi i]}$$

The $\otimes$-closure, i.e. internal hom $[-,-]$

For any two polynomials $p, q \in \mathsf{Poly}$, there is $[p, q] \in \mathsf{Poly}$ with

$$\mathsf{Poly}(p' \otimes p, q) \cong \mathsf{Poly}(p', [p, q])$$

for any $p' \in \mathsf{Poly}$. It can be given by the following formula:

$$[p, q] := \sum_{\varphi\colon p \to q} y^{\sum_{i \in p(1)} q[\varphi i]}$$

Let's examine it.

- A position of $[p, q]$ is a map $\varphi\colon p \to q$ of polynomials.
- What is a direction of $[p, q]$ at φ?
  - It's a pair (i, e) where $i \in p(1)$ and $e \in q[\varphi i]$.
  - We'll come back to this after we discuss coalgebras.

Properties of internal hom

The following are true of any internal hom, just written in Poly notation.

- $[y, p] \cong p$.
- $[p_1 \otimes p_2, p'] \cong [p_1, [p_2, p']]$.
- There is a map $p \otimes [p, q] \to q$ called *evaluation*. It induces:
  - A map $[p, q] \otimes [q, r] \to [p, r]$ called *internal composition*.
  - A map $[p_1, q_1] \otimes [p_2, q_2] \to [p_1 \otimes p_2, q_1 \otimes q_2]$ called *internal product*.

Later we will refer to these maps as the *standard maps.*

Coalgebras

A *coalgebra* for $F\colon \mathrm{Set} \to \mathrm{Set}$ is a set S and a map $S \to F(S)$.

- Let's refer to elements of S as *states*.
- For $p \in \mathrm{Poly}$ what does $f\colon S \to p(S)$ do to a state $s \in S$?
  - First it "reads out" a position $f^{\mathrm{rdt}}(s) \in p(1)$.
  - Then for each direction $d \in p[f^{\mathrm{rdt}}(s)]$, ...
  - ... it returns an "updated" state $f^{\mathrm{upd}}(s, d) \in S$.

Coalgebras

A *coalgebra* for $F\colon \mathrm{Set} \to \mathrm{Set}$ is a set S and a map $S \to F(S)$.

- Let's refer to elements of S as *states*.
- For $p \in \mathrm{Poly}$ what does $f\colon S \to p(S)$ do to a state $s \in S$?
  - First it "reads out" a position $f^{\mathrm{rdt}}(s) \in p(1)$.
  - Then for each direction $d \in p[f^{\mathrm{rdt}}(s)]$, ...
  - ... it returns an "updated" state $f^{\mathrm{upd}}(s, d) \in S$.

A *coalgebra map* $(S, f) \to (S', f')$ is a function $S \xrightarrow{g} S'$ with commuting

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[d, "g"'] & p(S) \arrow[d, "p(g)"] \\ S' \arrow[r, "f'"'] & p(S') \end{tikzcd}$$

This is very strong: any states $s \in S$ and $s' := g(s)$...

- ... have the same readout: $f'^{\mathrm{rdt}}(s') = f^{\mathrm{rdt}}(s)$, and...
- ... remain in sync after update: $f'^{\mathrm{upd}}(s', d) = g(f^{\mathrm{upd}}(s, d))$.
- That means they have the same observable behavior for all time.

Interaction patterns

Coalgebras can be tensored, giving $p\text{-Coalg} \times q\text{-Coalg} \to (p \otimes q)\text{-Coalg}$.

- Given coalgebras $f\colon S \to p(S)$ and $g\colon T \to q(T)$, we can form...
- ... their tensor product, $(f \otimes g)\colon (ST) \to (p \otimes q)(ST)$.
- It just runs an S-state and a T-state in parallel.

Interaction patterns

Coalgebras can be tensored, giving p-Coalg $\times$ q-Coalg $\to$ $(p \otimes q)$-Coalg.

- Given coalgebras $f\colon S \to p(S)$ and $g\colon T \to q(T)$, we can form...
- ... their tensor product, $(f \otimes g)\colon (ST) \to (p \otimes q)(ST)$.
- It just runs an S-state and a T-state in parallel.

Also, given a map $\varphi\colon p \to p'$, we get a map p-Coalg $\to$ p'-Coalg.

- φ translates a p-position readout to a p'-position readout.
- And it translates an incoming p'-direction back to a p-direction...
- ...which can be used to update the state.
- Mathematically, this is just the composite $S \xrightarrow{f} p(S) \xrightarrow{\varphi(S)} p'(S)$.

Interaction patterns

Coalgebras can be tensored, giving $p\text{-Coalg} \times q\text{-Coalg} \to (p \otimes q)\text{-Coalg}$.

- Given coalgebras $f\colon S \to p(S)$ and $g\colon T \to q(T)$, we can form...
- ... their tensor product, $(f \otimes g)\colon (ST) \to (p \otimes q)(ST)$.
- It just runs an S-state and a T-state in parallel.

Also, given a map $\varphi\colon p \to p'$, we get a map $p\text{-Coalg} \to p'\text{-Coalg}$.

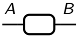
- φ translates a p-position readout to a p'-position readout.
- And it translates an incoming p'-direction back to a p-direction...
- ...which can be used to update the state.
- Mathematically, this is just the composite $S \xrightarrow{f} p(S) \xrightarrow{\varphi(S)} p'(S)$.

Together it means that given $p_1 \otimes \cdots \otimes p_k \xrightarrow{\varphi} p'$, we get

$$p_1\text{-Coalg} \times \cdots \times p_k\text{-Coalg} \to p'\text{-Coalg}$$

An interaction pattern φ takes dynamical systems in the p_i's to one in p'.

Interaction pattern example: wiring diagrams

Let's draw By^A as 

suggesting that it outputs B's and inputs A's.

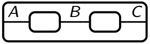
- Wiring diagrams are (special kinds of) interactions.
- Wiring rules: wires can split, but can't "pass".

Interaction pattern example: wiring diagrams

Let's draw By^A as 

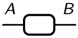
suggesting that it outputs B's and inputs A's.

- Wiring diagrams are (special kinds of) interactions.
- Wiring rules: wires can split, but can't "pass".



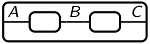
This shows a map $By^A \otimes Cy^B \to Cy^A$. Which one?

Interaction pattern example: wiring diagrams

Let's draw By^A as 

suggesting that it outputs B's and inputs A's.

- Wiring diagrams are (special kinds of) interactions.
- Wiring rules: wires can split, but can't "pass".



This shows a map $By^A \otimes Cy^B \to Cy^A$. Which one?

- $\mathsf{Poly}(BCy^{AB}, Cy^A) \cong \mathsf{Set}(BC, C) \times \mathsf{Set}(BCA, AB)$
- Outside world doesn't see B so we project it away.
- Inside boxes don't need C, so we project it away.
- Wiring diagrams are interactions with only projections/duplications.

Interactions patterns can be much more general, but this gives intuition.

The category $\mathcal{C}_p$ of p-trees

Let p be a polynomial. Define a *p-tree* to be...

- a rooted, possibly infinite tree, where ...
- every node is labeled with some $i \in p(1)$ and...
- has precisely $p[i]$-many branches coming out of it.

The category $\mathcal{C}_p$ of p-trees

Let p be a polynomial. Define a *p-tree* to be...

- a rooted, possibly infinite tree, where ...
- every node is labeled with some $i \in p(1)$ and...
- has precisely $p[i]$-many branches coming out of it.

More terminology for a p-tree T:

- A *root path* is a (finite) path f from the root to another node in T.
- At the other node sits a p-tree; we call this the *codomain of f*.

The category $\mathcal{C}_p$ of p-trees

Let p be a polynomial. Define a *p-tree* to be...

- a rooted, possibly infinite tree, where ...
- every node is labeled with some $i \in p(1)$ and...
- has precisely $p[i]$-many branches coming out of it.

More terminology for a p-tree T:

- A *root path* is a (finite) path f from the root to another node in T.
- At the other node sits a p-tree; we call this the *codomain of f*.

Define the *category of p-trees*, denoted $\mathcal{C}_p$:

- its objects are p-trees,
- a morphism $T \to U$ is a root-path in T with codomain U,
- composition is free; i.e. $\mathcal{C}_p$ is free on a graph.

The category $\mathcal{C}_p$ of p-trees

Let p be a polynomial. Define a *p-tree* to be...

- a rooted, possibly infinite tree, where ...
- every node is labeled with some $i \in p(1)$ and...
- has precisely $p[i]$-many branches coming out of it.

More terminology for a p-tree T:

- A *root path* is a (finite) path f from the root to another node in T.
- At the other node sits a p-tree; we call this the *codomain of f*.

Define the *category of p-trees*, denoted $\mathcal{C}_p$:

- its objects are p-trees,
- a morphism $T \to U$ is a root-path in T with codomain U,
- composition is free; i.e. $\mathcal{C}_p$ is free on a graph.

Facts:

- $\mathrm{Ob}(\mathcal{C}_p) \in \mathsf{Set}$ is the terminal p-coalgebra (use root and codomain)
- More generally, there is an equivalence p-$\mathsf{Coalg} \cong \mathcal{C}_p$-$\mathsf{Set}$.

Type theory and logic in the topos p-Coalg

Since p-Coalg $\cong \mathcal{C}_p$-Set, we know that p-coalgebras form a topos.

- That means you can do dependent type theory and higher-order logic.
- Example: $\underline{\mathbb{N}} = \{\phi : \mathbb{N} \to \texttt{Prop} \mid n \leq n' \Rightarrow \phi n' \Rightarrow \phi n\}$ = type of p-trees equipped with nondecreasing sequences of naturals.

Type theory and logic in the topos p-Coalg

Since p-Coalg $\cong \mathcal{C}_p$-Set, we know that p-coalgebras form a topos.

- That means you can do dependent type theory and higher-order logic.
- Example: $\underline{\mathbb{N}} = \{\phi : \mathbb{N} \to \texttt{Prop} \mid n \leq n' \Rightarrow \phi n' \Rightarrow \phi n\}$ = type of p-trees equipped with nondecreasing sequences of naturals.

A *logical proposition* means a subobject of the terminal object.

- So it is a collection ϕ of p-trees with the property that...
- ... if $T \in \phi$ then so is any p-subtree.

Logical operators and modalities:

- $\perp$ is empty, $\top$ is everything, $\vee$ is union, $\wedge$ is intersection.
- $\neg\phi$ is the set of p-trees that have no p-subtrees in ϕ.
- $\neg\neg\phi$ is "capable of ϕ": there's always a way to bring ϕ about.

Characters

How can we think about a logical proposition in p-Coalg?

- It's like a character, say "you" that's inhabiting the body p.
- You can express any of $p(1)$-many positions, and...
- ... for each $i \in p(1)$, there are $p[i]$ many things you could see happen.
- Your character is all the ways you might respond to what happens.
- (If we use all types, not just propositions, we get frequencies too.)

Example and non-example characters

So for example, here are some characters in $p = \{\bullet, \bullet, \bullet\}y^{\{\bullet, \bullet\}} \cong 3y^2$:

- "I'll output only green and red forever, any way I want."
- "I'll either output red forever or green forever."
- "I'll never output the same thing twice in a row".
- "I'll always output something different than I input".
- "In finite time I'll get to a point where I never output red again."
- "My output will never depend on the last three inputs received."
- "Whenever I receive two reds in a row, I'll do arbitrary stuff for three turns, then output red-blue-red-blue-$\cdots$ until I get a blue."
- "I'll never get locked out of ($\neg\neg$) responding tit-for-tat".

Here are some things you can't say (transients):

- "My first output is red."
- "For the next three turns I'll output only reds and blues".

Question: something like i.i.d., but what?

Consider the interface $p = \{H, T\}y \cong 2y$.

- A p-tree is just an infinite stream of H's and T's.
- If X is such a stream, it's not random: all its values are known.
- And yet, $[T, T, F, T, F, F, F, F, T, T, ...]$ has no discernable pattern.

Question: something like i.i.d., but what?

Consider the interface $p = \{H, T\}y \cong 2y$.

- A p-tree is just an infinite stream of H's and T's.
- If X is such a stream, it's not random: all its values are known.
- And yet, $[T, T, F, T, F, F, F, F, T, T, ...]$ has no discernable pattern.

What word ϕ from statistics am I looking for? "Normal"?

- And then let's check that ϕ is a proposition:
- if stream X satisfies ϕ, then its tail does too, right?

Question: something like i.i.d., but what?

Consider the interface $p = \{H, T\}y \cong 2y$.

- A p-tree is just an infinite stream of H's and T's.
- If X is such a stream, it's not random: all its values are known.
- And yet, $[T, T, F, T, F, F, F, F, T, T, ...]$ has no discernable pattern.

What word ϕ from statistics am I looking for? "Normal"?

- And then let's check that ϕ is a proposition:
- if stream X satisfies ϕ, then its tail does too, right?

What's the analogue for an arbitrary probability (Bernoulli) distribution?

Outline

Backprop as functor

Consider the compositional structure of deep learning.

- Each learner / neuron has inputs and outputs, say A and B.
- It has a parameter space, say S and an implementation function

$$I\colon S \times A \to B.$$

- Given a training pair $(a, b) \in A \times B$, it updates the parameter

$$U\colon S \times A \times B \to S$$

- ... and back-propagates some additional error to the input A:

$$R\colon S \times A \times B \to A.$$

Learners can be composed; the R map is indispensable.

Learners are coalgebras

An (A, B) learner is exactly a $[Ay^A, By^B]$-coalgebra!

$$[Ay^A, By^B] \cong \sum_{\varphi\colon Ay^A \to By^B} y^{AB}$$

So a map $f\colon S \to [Ay^A, By^B](S)$ assigns to each $s \in S$:

- a φ, i.e. a function $A \to B$ and a function $A \times B \to A$ (that's I and R)
- and a function $A \times B \to S$ (that's U).

Learners are coalgebras

An (A, B) learner is exactly a $[Ay^A, By^B]$-coalgebra!

$$[Ay^A, By^B] \cong \sum_{\varphi\colon Ay^A \to By^B} y^{AB}$$

So a map $f\colon S \to [Ay^A, By^B](S)$ assigns to each $s \in S$:

- a φ, i.e. a function $A \to B$ and a function $A \times B \to A$ (that's I and R)
- and a function $A \times B \to S$ (that's U).

So what does a learner do as a machine?

- The state of the learner is read out as a position in $[Ay^A, By^B]$,
- ... i.e. a function $A \to B$ and a function $A \times B \to A$ (a "lens").
- It receives a training pair $(a, b) \in A \times B$, causing state to update.

Learners are coalgebras

An (A, B) learner is exactly a $[Ay^A, By^B]$-coalgebra!

$$[Ay^A, By^B] \cong \sum_{\varphi\colon Ay^A \to By^B} y^{AB}$$

So a map $f\colon S \to [Ay^A, By^B](S)$ assigns to each $s \in S$:

- a φ, i.e. a function $A \to B$ and a function $A \times B \to A$ (that's I and R)
- and a function $A \times B \to S$ (that's U).

So what does a learner do as a machine?

- The state of the learner is read out as a position in $[Ay^A, By^B]$,
- ... i.e. a function $A \to B$ and a function $A \times B \to A$ (a "lens").
- It receives a training pair $(a, b) \in A \times B$, causing state to update.

These form a topos, and in the topos for $[\mathbb{R}^m y^{\mathbb{R}^m}, \mathbb{R}^n y^{\mathbb{R}^n}]$.

- "I'm smooth and do gradient descent and backprop" is a proposition.
- You could do more complex data science workflows in this language.

Toward $\mathcal{O}rg$

Let's look at a standard case, a $(\mathbb{R}^n, \mathbb{R})$-learner.

- Note that $\mathbb{R}^n y^{\mathbb{R}^n} \cong \mathbb{R}y^{\mathbb{R}} \otimes \cdots \otimes \mathbb{R}y^{\mathbb{R}}$.
- So these learners are coalgebras on $[\mathbb{R}y^{\mathbb{R}} \otimes \cdots \otimes \mathbb{R}y^{\mathbb{R}}, \mathbb{R}y^{\mathbb{R}}]$.

Let's abstract to coalgebras on $[p_1 \otimes \cdots \otimes p_k, p']$.

- These are machines that read out interaction patterns φ.
  - The φ aggregates the positions of all p_i's to form a position of p'
  - ... and takes a response from p' and distributes it to all the p_i's.
- An input to the coalgebra is just such an event:
  - It's a position of each p_i and a response from the environment p'.
  - The coalgebra takes that event and updates its state.

The categorical operad $\mathcal{O}rg$ of organizations

Define $\mathcal{O}rg$ to be the category-enriched operad with

- objects $\mathsf{Ob}(\mathcal{O}rg) = \mathsf{Ob}(\mathsf{Poly})$,
- morphisms $\mathcal{O}rg(p_1, \ldots, p_k; p') := [p_1 \otimes \cdots \otimes p_k, p']\text{-}\mathsf{Coalg}$
- with identity on p given by the $[p, p]$-coalgebra with one state,
- and composition given by the standard maps.

The categorical operad $\mathcal{O}rg$ of organizations

Define $\mathcal{O}rg$ to be the category-enriched operad with

- objects $\mathrm{Ob}(\mathcal{O}rg) = \mathrm{Ob}(\mathsf{Poly})$,
- morphisms $\mathcal{O}rg(p_1, \ldots, p_k; p') := [p_1 \otimes \cdots \otimes p_k, p']\text{-}\mathsf{Coalg}$
- with identity on p given by the $[p, p]$-coalgebra with one state,
- and composition given by the standard maps.

What does it mean?

- An object in $\mathcal{O}rg(p_1, \ldots, p_k; p')$ is like the officer of a company.
- She has resources $p_1, \ldots, p_k$ at her disposal.
- She determines how their output is shuttled internally,...
- ... and how it is delivered to the outside world p'.
- She receives a signal from the outside world and disperses it to the p_i.
- As this progresses, she changes her state, thus her approach.

Learners do this by gradient descent.

The $\mathcal{O}rg$-algebra of 0-ary morphisms

Consider the algebra $\mathcal{O}rg \to \mathsf{Cat}$ given by 0-ary morphisms.

- It sends $p \mapsto \mathcal{O}rg(; p) = [y, p]\text{-Coalg} \cong p\text{-Coalg}$.
- It sends an object $\varphi \in \mathcal{O}rg(p_1, \ldots, p_k; p')$ to the functor

$$p_1\text{-Coalg} \times \cdots \times p_k\text{-Coalg} \to p'\text{-Coalg}$$

given by tensoring and composing with φ as with wiring diagrams.

Let's call this the algebra of *dynamical systems*.

Fixed organizations

Consider the fixed points in $\mathcal{O}rg(p_1, \ldots, p_k; p')$.

- These are just (fixed) interaction patterns of the p_i's in p'.
- I.e., the way the p_i send and receive information is unchanging.
- A wiring diagram, e.g. of transistors in a computer, is fixed.
- There is a set-operad $\mathcal{O}rg_{fix}$ mapping to $\mathcal{O}rg$, ...
- ... and we can pullback the algebra of dynamical systems to it.

Fixed organizations

Consider the fixed points in $\mathcal{O}rg(p_1, \ldots, p_k; p')$.

- These are just (fixed) interaction patterns of the p_i's in p'.
- I.e., the way the p_i send and receive information is unchanging.
- A wiring diagram, e.g. of transistors in a computer, is fixed.
- There is a set-operad $\mathcal{O}rg_{fix}$ mapping to $\mathcal{O}rg$, ...
- ... and we can pullback the algebra of dynamical systems to it.

So the more general $\mathcal{O}rg$ is about adjusting interaction patterns.

You as an organization

Think of the resources at your disposal.

- This depends on how we define "you" but let's go with colloquial.
- Some resources: eyes, limbs, thoughts, heart, car, bank account.
- They respond to signals from each other and the outside world.

We might imagine you as a switchboard operator.

- When X comes into your eyes, you try not to think about it.
- When your heart rate increases, you decide to call the bank.

You control the relays, though you don't control how they respond.

- The bank may not send the check,
- Your limb may not work as expected.
- But you are cognizant of what they're doing.

If the way you control the relays can change through time, congrats!

Gambling games in $\mathbb{O}rg$

Let's consider "finite state gamblers" in $\mathbb{O}rg$.

- Let $[N] := \{1, \ldots, N\}$ and $\Delta_N := \{p \colon [N] \to \mathbb{R}_{\geq 0} \mid \sum_i p_i = N\}$
- Think of $p \in \Delta_N$ as a *bet* on an N-sided die.
  - For example, suppose $N = 2$ and $p(1) = .6$ and $p(2) = 1.4$.
  - If the die turns up 1, we multiply your wealth by 0.6.
  - If the die turns up 2, we multiply your wealth by 1.4.
- Let $m_N := \Delta_N y^N$. A *gambler on alphabet N* is an m_N-coalgebra.
  - That is, it's a machine that produces bets and receives die rolls.

Gambling games in $\mathbb{O}\mathfrak{rg}$

Let's consider "finite state gamblers" in $\mathbb{O}\mathfrak{rg}$.

- Let $[N] := \{1, \ldots, N\}$ and $\Delta_N := \{p \colon [N] \to \mathbb{R}_{\geq 0} \mid \sum_i p_i = N\}$
- Think of $p \in \Delta_N$ as a *bet* on an N-sided die.
  - For example, suppose $N = 2$ and $p(1) = .6$ and $p(2) = 1.4$.
  - If the die turns up 1, we multiply your wealth by 0.6.
  - If the die turns up 2, we multiply your wealth by 1.4.
- Let $m_N := \Delta_N y^N$. A *gambler on alphabet N* is an m_N-coalgebra.
  - That is, it's a machine that produces bets and receives die rolls.

The game itself is an object in $\mathbb{O}\mathfrak{rg}(Ny, m_N; y)$.

- That is, it's a $[Ny \otimes \Delta_N y^N, y]$-coalgebra.
- Its state set is $\mathbb{R}$; that is, the officer keeps track of the bettor's wealth.
- The map $\mathbb{R} \to [Ny \otimes \Delta_N y^N, y](\mathbb{R})$ is given by

$$w \mapsto (i, p) \mapsto (i, w * p(i))$$

The notation is hard to decipher, but it says that the officer sends the outcome i to the bettor and tracks the wealth as above.

Outline

How I'm approaching this

The above was done with two competing objectives.

- To follow the math, so that it remains elegant and builds on itself.
  - Poly is an abundant category, with great formal properties.
  - $\mathcal{O}\mathfrak{r}g$ is just packaging up coalgebra interactions.
  - $\mathcal{O}\mathfrak{r}g$ generalizes "backprop as functor"...
  - ... and wiring diagrams of dynamical systems.
  - I can study its "behavior types" with topos theory.
- To ask that the math follow aspects of life and experience.
  - Learners loosely model neurons in the brain.
  - Martingales model betting, which has deep biological roots.
  - A company officer directs the interaction between resources.

What makes us "good" at this stuff: learning, betting, officiating?

- Here we are, communicating. What makes it "work"?

Flow and ping

People often talk about flow states as pleasurable and effective.

- If we look at the interactions between your subsystems, ...
- ... what would we see, especially during flow?
- Is there some notion of "efficiency" there?

Flow and ping

People often talk about flow states as pleasurable and effective.

- If we look at the interactions between your subsystems, ...
- ... what would we see, especially during flow?
- Is there some notion of "efficiency" there?

In golf, baseball, or tennis, a good hit often comes with a "ping".

- In this case, you barely feel the ball hit your club, bat, or racket,...
- ... and yet the ball goes flying.
- In contrast, when you "flub" it, you get pain and no distance.
- So what is that ping? Is there some notion of "efficiency" there?
  - If we slow ping down by 1000x, would we see flow?
  - Does an hour of flow look like a ping when we look back on it?
- As an officer of an organization, how can you adjust to make ping?
- Can we say what "collective cognition" is?

Flow and ping

People often talk about flow states as pleasurable and effective.

- If we look at the interactions between your subsystems, ...
- ... what would we see, especially during flow?
- Is there some notion of "efficiency" there?

In golf, baseball, or tennis, a good hit often comes with a "ping".

- In this case, you barely feel the ball hit your club, bat, or racket,...
- ... and yet the ball goes flying.
- In contrast, when you "flub" it, you get pain and no distance.
- So what is that ping? Is there some notion of "efficiency" there?
  - If we slow ping down by 1000x, would we see flow?
  - Does an hour of flow look like a ping when we look back on it?
- As an officer of an organization, how can you adjust to make ping?
- Can we say what "collective cognition" is?

Is there math for flow or ping?

Rayleigh-Bénard convection

Patterns can form when there is a potential difference across a medium.

- Rayleigh-Bénard convection is one such example.
  - Given a temp. difference across a fluid (e.g. bottom is hotter)...
  - ... you get structure: a bunch of rotating "cells".
  - They increase the rate at which the temperature equalizes.
- A hurricane's eye efficiently moves heat from ocean to atmosphere.

Rayleigh-Bénard convection

Patterns can form when there is a potential difference across a medium.

- Rayleigh-Bénard convection is one such example.
  - Given a temp. difference across a fluid (e.g. bottom is hotter)...
  - ... you get structure: a bunch of rotating "cells".
  - They increase the rate at which the temperature equalizes.
- A hurricane's eye efficiently moves heat from ocean to atmosphere.

Some believe that life is such a pattern, designed to disperse entropy.

- Can anything like this be seen from the $\mathcal{O}\mathfrak{r}g$ point of view?
  - E.g., you might want to receive relatively predictable inputs...
  - ... while producing relatively unpredictable outputs.
- Can we say that sort of thing?
- Can we setup simplified physics and see Rayleigh-Bénard convection?

Sense-making

Let's say our goal is sense-making.

- We can think of this as channeling or processing negentropy.
- We channel it to the systems that sustain us.
  - We get "food" to our organs to process.
  - We seek out information that our brains can make sense of.
  - We tell our friends of useful things we find out.

Sense-making

Let's say our goal is sense-making.

- We can think of this as channeling or processing negentropy.
- We channel it to the systems that sustain us.
  - We get "food" to our organs to process.
  - We seek out information that our brains can make sense of.
  - We tell our friends of useful things we find out.

This should be its own reward: it structures and enables us.

- I have a feeling that this can lead to natural economics.
- So how do you see it in the math?
- In particular, I imagine it should make sense in $\mathcal{O}\mathfrak{r}g$, but how?

Sense-making

Let's say our goal is sense-making.

- We can think of this as channeling or processing negentropy.
- We channel it to the systems that sustain us.
  - We get "food" to our organs to process.
  - We seek out information that our brains can make sense of.
  - We tell our friends of useful things we find out.

This should be its own reward: it structures and enables us.

- I have a feeling that this can lead to natural economics.
- So how do you see it in the math?
- In particular, I imagine it should make sense in $\mathcal{O}\mathfrak{r}g$, but how?

So that's where I'm stuck. Can you help me make sense of it?

Overall fit

How does all this resonate with you?

- Comments?
- Questions?
- Ideas on where to go?