

# Triple categories of open cybernetic systems

Matteo Capucci

MSP group, University of Strathclyde

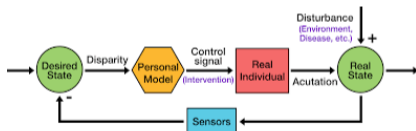
ItaCa Fest 2022

October 18th, 2022

# Introduction

Cybernetic systems are

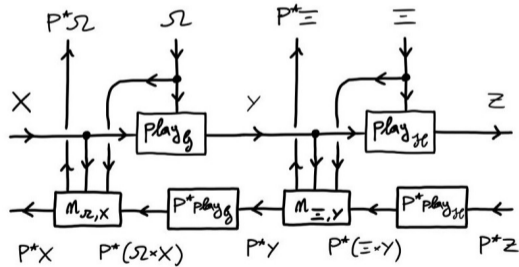
systems with control mechanisms and interactive feedback



# Introduction

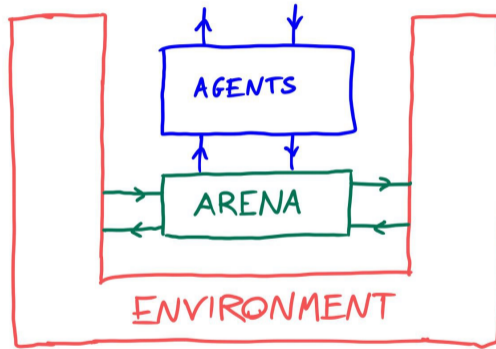
Cybernetic systems are

systems with control mechanisms and interactive feedback



## Introduction

As systems, they have a very peculiar structure:



where **agents** = controllers, players, learners, species, genes, etc.

and **arena** = physical system, computer program, game play, ecosystem, etc.

## Introduction

The goal of categorical cybernetics is to develop adequate categorical abstractions for

1. formulating models of cybernetic systems,
2. proving properties of general cybernetic systems (e.g. FEP, universality, etc.)
3. simulating and study such models,
4. writing **effective**, **efficient** and **effable** software to do so.

## Introduction

The most interesting part of a cybernetic system is the **controller** sub-system, which (mostly) decides how the rest of the system behaves. In particular, that's usually where the 'new' stuff happens, otherwise we'd be studying dynamical systems.

## Introduction

The most interesting part of a cybernetic system is the **controller** sub-system, which (mostly) decides how the rest of the system behaves. In particular, that's usually where the 'new' stuff happens, otherwise we'd be studying dynamical systems.

Thus the categorical study of cybernetics mostly deals with

1. understanding the structure of control mechanisms of a given kind,
2. understanding their compositional characteristic,
3. **measuring and predicting their behaviour.**

## Introduction

The most interesting part of a cybernetic system is the **controller** sub-system, which (mostly) decides how the rest of the system behaves. In particular, that's usually where the 'new' stuff happens, otherwise we'd be studying dynamical systems.

Thus the categorical study of cybernetics mostly deals with

1. understanding the structure of control mechanisms of a given kind,
2. understanding their compositional characteristic,
3. **measuring and predicting their behaviour.**

Currently, we know how to do this best for **categorical game theory** [Cap22], **categorical machine learning** [Cru+21] and a bit more [Cap+21a; Smi21].

## Plan

Today I will talk about a nascent framework to systematize our research and leverage the current knowledge to open up new applicability frontiers.

## Plan

Today I will talk about a nascent framework to systematize our research and leverage the current knowledge to open up new applicability frontiers.

It is grounded in David Jaz Myers' framework for categorical system theory [Mye22]. Categorical system theory frames behavioural system theory as the study/construction of maps

$$\mathbb{S}\text{ys} \xrightarrow{B} \mathbb{O}\text{bs}$$

<b>specification</b>	<b>behaviour</b>
(syntax)	(semantics)

The first half of the talk is almost completely dedicated to introducing this framework.

## Plan

Today I will talk about a nascent framework to systematize our research and leverage the current knowledge to open up new applicability frontiers.

It is grounded in David Jaz Myers' framework for categorical system theory [Mye22]. Categorical system theory frames behavioural system theory as the study/construction of maps

$$\mathbb{S}\text{ys} \xrightarrow{B} \mathbb{O}\text{bs}$$

<b>specification</b>	<b>behaviour</b>
(syntax)	(semantics)

The first half of the talk is almost completely dedicated to introducing this framework.

I'll then move beyond to explain how triple categories enter the field when we move to the cybernetical settings. There I will give some motivating examples.

## Plan

Overall, I'm afraid I will leave you with more questions than answers. I hope you can help me to reverse this situation!

## Plan

Overall, I'm afraid I will leave you with more questions than answers. I hope you can help me to reverse this situation!

Many thanks to Jules Hedges, Bruno Gavranović, Dylan Braithwaite, Riu Rodríguez Sakamoto, Philipp Zahn, Manuel Baltieri, Brandon Shapiro, David Spivak, and David Jaz Myers for precious conversations about this project.

# Plan

Notational conventions in the talk:

1. double categories are weak by default,
2. loose arrows are vertical (denoted:  $\dashrightarrow$ ), tight arrows are horizontal (denoted:  $\rightarrow$ ),
3.  $\mathbf{Set}$  denotes the double category of spans in  $\mathbf{Set}$ ,
4.  $\mathbf{Cat}$  denotes the double category categories, functors and profunctors.

Also I diverged from [Mye22] on some notational and terminological choices.

# Categorical system theory

## Categorical system theory

Categorical system theory is a simple framework:

1. It starts with recognizing processes organize in double categories:

$\mathbb{P}$

## Categorical system theory

Categorical system theory is a simple framework:

1. It starts with recognizing processes organize in double categories:

$$\mathbb{P}$$

2. Then processes are used to index systems, giving rise to doubly indexed categories of systems:

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{normal lax}} \mathbf{Cat}$$

## Categorical system theory

Categorical system theory is a simple framework:

1. It starts with recognizing processes organize in double categories:

$$\mathbb{P}$$

2. Then processes are used to index systems, giving rise to doubly indexed categories of systems:

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{normal lax}} \mathbf{Cat}$$

3. Finally, behaviour is studied by describing maps into the *observational theory*:

$$\begin{array}{ccc} \mathbb{P}^{\top} & \xrightarrow{B^{\top}} & \mathbf{Set}^{\top} \\ \mathbf{Sys} \downarrow & \xrightarrow{B^b} & \downarrow \mathbf{Obs} \\ \mathbf{Cat} & \xlongequal{\quad} & \mathbf{Cat} \end{array}$$

## Process theories

### Definition

A **process theory** is a double category with attitude where:

# Process theories

## Definition

A **process theory** is a double category with attitude where:

1. objects are **boundaries** or **interfaces**,

# Process theories

## Definition

A **process theory** is a double category with attitude where:

1. objects are **boundaries** or **interfaces**,
2. vertical (loose) morphisms are **processes**,

# Process theories

## Definition

A **process theory** is a double category with attitude where:

1. objects are **boundaries** or **interfaces**,
2. vertical (loose) morphisms are **processes**,
3. horizontal (tight) morphisms are **maps of boundaries**,

## Process theories

### Definition

A **process theory** is a double category with attitude where:

1. objects are **boundaries** or **interfaces**,
2. vertical (loose) morphisms are **processes**,
3. horizontal (tight) morphisms are **maps of boundaries**,
4. squares are **maps of processes**

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ \downarrow p & \xRightarrow{\sigma} & \downarrow q \\ B & \xrightarrow{k} & D \end{array}$$

## Process theories

### Definition

A **process theory** is a double category with attitude where:

1. objects are **boundaries** or **interfaces**,
2. vertical (loose) morphisms are **processes**,
3. horizontal (tight) morphisms are **maps of boundaries**,
4. squares are **maps of processes**

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ \downarrow p & \xRightarrow{\sigma} & \downarrow q \\ B & \xrightarrow{k} & D \end{array}$$

One can define more refined kinds of process theories based on the structure of the double category, i.e. **concurrent process theories** have a monoidal product with the attitude of being juxtaposition of processes in space.

## Process theories

I think of boundaries as 'wires' that can transmit a certain kind of signal, possibly in many directions. Maps of boundaries are ways to translate between wires of different kinds, but they are **static**.

OTOH processes can be thought as wiring diagrams, they 'make things happen', i.e. they are **dynamical**.

Maps of processes are again **static**, they tell us how to interpret one process into another, mediated by an interpretation of the boundaries.

## Example: lenses

The process theory of **lenses**  $\mathbb{L}ens$  is defined as follows:

1. boundaries are pairs of sets  $\begin{pmatrix} A^- \\ A^+ \end{pmatrix}$ ,
2. processes are lenses:

$$\begin{pmatrix} A^- \\ A^+ \end{pmatrix} \begin{array}{c} \xleftarrow{p^\sharp} \\ \xrightarrow{p} \end{array} \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

where

$$p : A \rightarrow B, \quad p^\sharp : A^+ \times B^- \rightarrow A^-$$

3. maps of boundaries are charts:

$$\begin{pmatrix} A^- \\ A^+ \end{pmatrix} \begin{array}{c} \xrightarrow{h^b} \\ \xrightarrow{h} \end{array} \begin{pmatrix} C^- \\ C^+ \end{pmatrix}$$

where

$$h : A \rightarrow C, \quad h^b : A^+ \times A^- \rightarrow C^-$$

## Example: lenses

3. maps of processes are 'commutative squares':

$$\begin{array}{ccc} \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{h^b} \\ \xleftarrow{h} \end{array} & \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \\ p^\sharp \uparrow \downarrow p & & q^\sharp \uparrow \downarrow q \\ \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{k^b} \\ \xleftarrow{k} \end{array} & \begin{pmatrix} D^- \\ D^+ \end{pmatrix} \end{array}$$

Such an arrangement is a square iff for every  $a^+ \in A^+$  and  $b^- \in B^-$ :

$$\begin{aligned} q(h(a^+)) &= k(p(a^+)), \\ h^b(a^+, p^\sharp(a^+, b^-)) &= q^\sharp(h(a^+), k^b(p(a^+), b^-)). \end{aligned}$$

**We'll see it in action later! Hopefully it will make it clearer**

## Example: non-deterministic lenses

As a variation, the **process theory of non-deterministic lenses**  $\mathbb{L}ens_P$  is given by lenses and charts with an effect in their second part:

**lenses**

$$\left( \begin{array}{c} A^- \\ A^+ \end{array} \right) \begin{array}{c} \xleftarrow{p^\sharp} \\ \xrightarrow{p} \end{array} \left( \begin{array}{c} B^- \\ B^+ \end{array} \right)$$

$$p : A \rightarrow B$$

$$p^\sharp : A^+ \times B^- \rightarrow PA^-$$

**charts**

$$\left( \begin{array}{c} A^- \\ A^+ \end{array} \right) \begin{array}{c} \xrightarrow{h^b} \\ \xleftarrow{h} \end{array} \left( \begin{array}{c} C^- \\ C^+ \end{array} \right)$$

$$h : A \rightarrow C$$

$$h^b : A^+ \times A^- \rightarrow PC^-$$

In this case, the commutativity condition on square is easily relaxed:

$$q(h(a^+)) = k(p(a^+)),$$

$$h^b(a^+, p^\sharp(a^+, b^-)) \subseteq q^\sharp(h(a^+), k^b(p(a^+), b^-)).$$

## Example: dependent lenses

The **process theory of dependent lenses** is very similar to the previous but now lenses and charts are 'dependently typed'. We start from an indexed category  $F : \mathcal{C}^{\text{op}} \rightarrow \mathbf{Cat}$ , define boundaries  $\left( \begin{smallmatrix} A^- \\ A^+ \end{smallmatrix} \right)$  to be dependent pairs  $(A^+ : \mathcal{C}, A^- : FA^+)$ , and

**lenses**

$$\left( \begin{smallmatrix} A^- \\ A^+ \end{smallmatrix} \right) \begin{array}{c} \xleftarrow{p^\sharp} \\ \xrightarrow{p} \end{array} \left( \begin{smallmatrix} B^- \\ B^+ \end{smallmatrix} \right)$$

$$p : A \rightarrow B$$

$$p^\sharp : (a^+ : A^+) \times B^-(f(a^+)) \rightarrow A^-(a)$$

**charts**

$$\left( \begin{smallmatrix} A^- \\ A^+ \end{smallmatrix} \right) \begin{array}{c} \xrightarrow{h^\flat} \\ \xleftarrow{h} \end{array} \left( \begin{smallmatrix} C^- \\ C^+ \end{smallmatrix} \right)$$

$$h : A \rightarrow C$$

$$h^\flat : (a^+ : A^+) \times A^-(a^+) \rightarrow C^-(h(a^+))$$

## Example

Let  $\text{Sub}/- : \mathbf{Smooth}^{\text{op}} \rightarrow \mathbf{Cat}$  be the submersion slice, i.e.  $\text{Sub}/X$  is the subcat of  $\mathbf{Smooth}/X$  given by submersions. Then we can arrange smooth manifolds and submersions in the process theory

$\mathbb{L}\text{ens}_{\text{Sub}}$  of **smooth processes**.

## Example: observational theories

If  $\mathcal{C}$  has pullbacks,  $\mathbb{S}\text{pan}(\mathcal{C})$  is the **observational theory of  $\mathcal{C}$ -processes** where

1. boundaries are objects in  $\mathcal{C}$ ,
2. processes are spans in  $\mathcal{C}$ ,
3. maps of boundaries are maps in  $\mathcal{C}$ ,
4. maps of processes are squares in  $\mathbb{S}\text{pan}(\mathcal{C})$ :

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ p_\ell \uparrow & & \uparrow q_\ell \\ S & \xrightarrow{\sigma} & R \\ p_r \downarrow & & \downarrow q_r \\ B & \xrightarrow{k} & D \end{array}$$

This is an *observational theory*: processes are described by what we observe about their internal states.

This the archetypal form of **behaviour**: in fact, any process theory admits (horizontal) hom-functors into it, mapping a boundary  $C$  to the set of its observations  $\mathbb{P}^h(A, C)$  of type  $A$ .

## Systems vs processes

We have setup a way to talk about processes and maps thereof, but in practice we often care about specific processes, namely **stateful/one-sided** ones.

## Systems vs processes

We have setup a way to talk about processes and maps thereof, but in practice we often care about specific processes, namely **stateful/one-sided** ones.

### Example

Among (non-deterministic) lenses, stateful ones are known as **(non-deterministic) Moore machines**:

$$\left( \begin{array}{c} \text{update} \\ \text{output} \end{array} \right) : \left( \begin{array}{c} S \\ S \end{array} \right) \rightleftarrows \left( \begin{array}{c} I \\ O \end{array} \right)$$

This is an interactive finite state machine.

## Systems vs processes

We have setup a way to talk about processes and maps thereof, but in practice we often care about specific processes, namely **stateful/one-sided** ones.

### Example

Among all maps of smooth manifolds, we especially care about **(non-autonomous) ODEs**:

$$\begin{pmatrix} \text{update} \\ \text{expose} \end{pmatrix} : \begin{pmatrix} TS \\ S \end{pmatrix} \rightleftarrows \begin{pmatrix} I \\ O \end{pmatrix}$$

This models the evolution of a quantity  $\text{expose} : S \rightarrow O$  on a state space  $S$ , influenced by a control  $i$ , under the rule  $\dot{s} = \text{update}(s, i)$ .

## Systems vs processes

We have setup a way to talk about processes and maps thereof, but in practice we often care about specific processes, namely **stateful/one-sided** ones.

### Example

Among all spans, we can consider 'one-sided' ones as state spaces exposing some observables:

$$\text{observe} : S \rightarrow O$$

## System theories

Crucially, **systems** are

1. Rewirable along processes that 'extend' them, changing their interfaces,
2. Comparable with other systems given a way to compare their interfaces and states

## System theories

Crucially, **systems** are

1. Rewirable along processes that 'extend' them, changing their interfaces,
2. Comparable with other systems given a way to compare their interfaces and states

### Definition

A **system theory** over the process theory  $\mathbb{P}$  is a doubly indexed category with attitude, i.e. a normal lax double functor\*

$$\mathbf{Sys} : \mathbb{P}^{\mathbf{T}} \longrightarrow \mathbf{Cat}$$

\*this is better defined as a 'left  $\mathbb{P}$ -module' in  $\mathbf{Span}(\mathbf{Cat})$ , since we are exchanging loose and tight arrows (thanks to DJM for clarifying this).

## System theories

The attitude for a system theory is as follows:

## System theories

The attitude for a system theory is as follows:

1. the categories  $\mathbb{S}\text{ys}(A)$  for a given boundary  $A : \mathbb{P}$  are categories of **systems with boundary  $A$**  and **simulations\*** thereof,

\*These are not exactly simulations in the coalgebraic sense, but conceptually they are.

## System theories

The attitude for a system theory is as follows:

1. the categories  $\mathbb{S}\text{ys}(A)$  for a given boundary  $A : \mathbb{P}$  are categories of **systems with boundary  $A$**  and **simulations\*** thereof,
2. processes  $p : A \dashrightarrow B$  in  $\mathbb{P}$  act functorially by **rewiring**:

$$\mathbb{S}\text{ys}(p) : \mathbb{S}\text{ys}(A) \longrightarrow \mathbb{S}\text{ys}(B)$$

## System theories

The attitude for a system theory is as follows:

1. the categories  $\mathbb{S}\text{ys}(A)$  for a given boundary  $A : \mathbb{P}$  are categories of **systems with boundary  $A$  and simulations\*** thereof,
2. processes  $p : A \dashrightarrow B$  in  $\mathbb{P}$  act functorially by **rewiring**:

$$\mathbb{S}\text{ys}(p) : \mathbb{S}\text{ys}(A) \longrightarrow \mathbb{S}\text{ys}(B)$$

3. maps of boundaries  $h : A \rightarrow C$  act profunctorially (and laxly!):

$$\mathbb{S}\text{ys}(h) : \mathbb{S}\text{ys}(A) \dashrightarrow \mathbb{S}\text{ys}(C)$$

and we think of  $\mathbb{S}\text{ys}(h)(S, R)$  as the **possible simulations of  $S : \mathbb{S}\text{ys}(A)$  in  $R : \mathbb{S}\text{ys}(C)$  mediated by the map  $h$  on their boundaries.**

# System theories

## 4. maps of processes

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ p \bullet \downarrow & \xRightarrow{\sigma} & \bullet \downarrow q \\ B & \xrightarrow{k} & D \end{array}$$

yield squares

$$\begin{array}{ccc} \mathbf{Sys}(A) & \xrightarrow{\mathbf{Sys}(p)} & \mathbf{Sys}(B) \\ \mathbf{Sys}(h) \downarrow & \xRightarrow{\mathbf{Sys}(\sigma)} & \downarrow \mathbf{Sys}(k) \\ \mathbf{Sys}(C) & \xrightarrow{\mathbf{Sys}(q)} & \mathbf{Sys}(D) \end{array}$$

which **extend** a given simulation of systems along the given map of systems  $\sigma$ .

## Example: Moore machines

Moore machines are (one possible notion of) **system** associated to the process theory  $\mathbb{L}ens$ :

$$\mathbf{Moore}\left(\begin{array}{c} I \\ O \end{array}\right) = \left\{ S \xrightarrow{g} R \left| \begin{array}{ccc} \begin{array}{c} (S) \\ (S) \end{array} & \xrightarrow[\quad g \quad]{\quad \pi_2 \circ g \quad} & \begin{array}{c} (R) \\ (R) \end{array} \\ \begin{array}{c} \uparrow \text{update}_S \\ \downarrow \text{output}_S \end{array} & & \begin{array}{c} \uparrow \text{update}_R \\ \downarrow \text{output}_R \end{array} \\ \begin{array}{c} (I) \\ (O) \end{array} & \xlongequal{\quad} & \begin{array}{c} (I) \\ (O) \end{array} \end{array} \right\}$$

A Moore machine  $S : \mathbf{Moore}\left(\begin{array}{c} I \\ O \end{array}\right)$  is a lens whose domain is a specified set of *states*  $S$ . Its other boundary is just an input-output boundary  $\left(\begin{array}{c} I \\ O \end{array}\right)$ .

A morphism  $g : S \rightarrow R$  in  $\mathbf{Moore}\left(\begin{array}{c} I \\ O \end{array}\right)$  is a way to interpret the execution of  $S$  in the execution of  $R$  by mapping states. Crucially, we restrict the chart on states to have the form  $(\pi_2 \circ g, g)$ , i.e. to be the same on both levels.

## Example: Moore machines

### Example

Call  $\text{fix}_{\bar{o}}$  the MM with a single state and output  $\text{fix}_{\bar{o}} = 1 \xrightarrow{\bar{o}} O$ . Consider a morphism  $\text{fix}_{\bar{o}} \xrightarrow{r} R$ :

$$\begin{array}{ccc} \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{array}{c} \xrightarrow{\pi_2 \circ r} \\ \xrightarrow{r} \end{array} & \begin{pmatrix} R \\ R \end{pmatrix} \\ \begin{array}{c} \uparrow \text{update}_{\text{fix}_{\bar{o}}} \\ \downarrow \text{output}_{\text{fix}_{\bar{o}}} \end{array} & & \begin{array}{c} \uparrow \text{update}_R \\ \downarrow \text{output}_R \end{array} \\ \begin{pmatrix} I \\ O \end{pmatrix} & \xlongequal{\quad} & \begin{pmatrix} I \\ O \end{pmatrix} \end{array}$$

The 'commutativity' condition of such a square in  $\mathbb{L}\text{ens}$  imposes:

$$\text{output}_R(r) = \bar{o}, \quad \forall i \in I, \text{update}_R(i, r) = r$$

This map **picks out a state in R that behaves like**  $\text{fix}_{\bar{o}}$ :

$r$  is a steady state constantly emitting  $\bar{o} \in O$ .

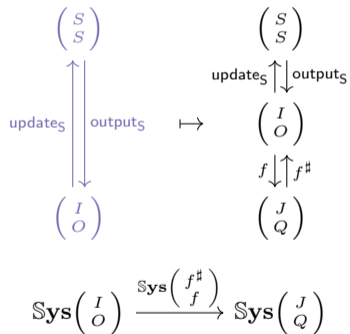
## Example: Moore machines

Moore machines can be rewired along their boundaries by lenses (processes in `Moore`), yielding a new Moore machine:

$$\begin{array}{ccc} \begin{pmatrix} S \\ S \end{pmatrix} & & \\ \uparrow \text{updates}_S & \parallel & \text{outputs}_S \downarrow \\ \begin{pmatrix} I \\ O \end{pmatrix} & \begin{array}{c} \xleftarrow{f^\sharp} \\ \xrightarrow{f} \end{array} & \begin{pmatrix} J \\ Q \end{pmatrix} \end{array}$$

## Example: Moore machines

Moore machines can be rewired along their boundaries by lenses (processes in `Moore`), yielding a new Moore machine:



## Example: Moore machines

Given a chart  $\begin{pmatrix} k^b \\ k \end{pmatrix} : \begin{pmatrix} I \\ O \end{pmatrix} \Rightarrow \begin{pmatrix} J \\ Q \end{pmatrix}$  we get a profunctor:

$$\mathbf{Moore}\left(\begin{pmatrix} I \\ O \end{pmatrix}\right)^{\text{op}} \times \mathbf{Moore}\left(\begin{pmatrix} J \\ Q \end{pmatrix}\right) \xrightarrow{\mathbf{Moore}\left(\begin{pmatrix} k^b \\ k \end{pmatrix}\right)} \mathbf{Set}$$

$$\begin{array}{ccc} \begin{array}{c} \begin{pmatrix} S \\ S \end{pmatrix} \\ \updownarrow \text{update}_S \text{ / } \text{output}_S \\ \begin{pmatrix} I \\ O \end{pmatrix} \end{array} & , & \begin{array}{c} \begin{pmatrix} R \\ R \end{pmatrix} \\ \updownarrow \text{update}_R \text{ / } \text{output}_R \\ \begin{pmatrix} J \\ Q \end{pmatrix} \end{array} \end{array} \mapsto \left\{ \begin{array}{ccc} \begin{pmatrix} S \\ S \end{pmatrix} & \begin{array}{c} \xrightarrow{\pi_2 \circ h} \\ \xrightarrow{h} \end{array} & \begin{pmatrix} R \\ R \end{pmatrix} \\ \updownarrow \text{update}_S \text{ / } \text{output}_S & & \updownarrow \text{update}_R \text{ / } \text{output}_R \\ \begin{pmatrix} I \\ O \end{pmatrix} & \begin{array}{c} \xrightarrow{k^b} \\ \xrightarrow{k} \end{array} & \begin{pmatrix} J \\ Q \end{pmatrix} \end{array} \right\}$$

These start to be interesting: they talk about systems with possibly different interfaces.

## Example: Moore machines

### Example

Consider  $\text{Clock} : \text{Moore}\left(\frac{1}{\mathbb{N}}\right)$ , the Moore machine with state space  $\mathbb{N}$  defined as:

$$\text{output}_{\text{Clock}} : \mathbb{N} \longrightarrow \mathbb{N} := 1_{\mathbb{N}},$$

$$\text{update}_{\text{Clock}} : \mathbb{N} \times 1 \longrightarrow \mathbb{N} := n \mapsto n + 1$$

Then for each a chart  $\left(\frac{1}{\mathbb{N}}\right) \xrightarrow[i]{o} \left(\frac{I}{O}\right)$ , the set  $\text{Moore}\left(\frac{i}{o}\right)(\text{Clock}, S)$  is the set of squares:

$$\begin{array}{ccc}
 \left(\frac{\mathbb{N}}{\mathbb{N}}\right) & \xrightarrow[\text{---}]{\pi_2 \circ s_n} \left(\frac{S}{S}\right) \\
 \uparrow \text{update}_{\text{Clock}} & & \uparrow \text{update}_S \\
 \left(\frac{1}{\mathbb{N}}\right) & \xrightarrow[\text{---}]{i_n} \left(\frac{I}{O}\right) \\
 \downarrow \text{output}_{\text{Clock}} & & \downarrow \text{output}_S
 \end{array}$$

$\text{output}_S(s_n) = o_n,$   
 $\text{update}_S(r_n, i_n) = r_{n+1}$

which means  $s : \mathbb{N} \rightarrow S$  is a **trajectory** of the machine  $S$ .

## Example: Moore machines

Finally, given a square

$$\begin{array}{ccc}
 \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{h^b} \\ \xleftarrow{h} \end{array} & \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \\
 p^\sharp \uparrow \downarrow p & & q^\sharp \uparrow \downarrow q \\
 \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{k^b} \\ \xleftarrow{k} \end{array} & \begin{pmatrix} D^- \\ D^+ \end{pmatrix}
 \end{array}$$

we get a square in  $\mathbb{C}\text{at}$ ...

$$\begin{array}{ccc}
 \text{Moore} \left( \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \right) & \xrightarrow{\text{Moore} \left( \begin{pmatrix} p^\sharp \\ p \end{pmatrix} \right)} & \text{Moore} \left( \begin{pmatrix} B^- \\ B^+ \end{pmatrix} \right) \\
 \text{Moore} \left( \begin{pmatrix} h^b \\ h \end{pmatrix} \right) \downarrow & \xrightarrow{\text{Moore}(\square)} & \downarrow \text{Moore} \left( \begin{pmatrix} k^b \\ k \end{pmatrix} \right) \\
 \text{Moore} \left( \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \right) & \xrightarrow{\text{Moore} \left( \begin{pmatrix} q^\sharp \\ q \end{pmatrix} \right)} & \text{Moore} \left( \begin{pmatrix} D^- \\ D^+ \end{pmatrix} \right)
 \end{array}$$

## Example: Moore machines

...given by stacking squares:

$$\text{Moore}\left(\begin{smallmatrix} h^b \\ h \end{smallmatrix}\right)(S, R) \xrightarrow{\text{Moore}(\square)_{S,R}} \text{Moore}\left(\begin{smallmatrix} k^b \\ k \end{smallmatrix}\right)\left(\text{Moore}\left(\begin{smallmatrix} p^\# \\ p \end{smallmatrix}\right)(S), \text{Moore}\left(\begin{smallmatrix} q^\# \\ q \end{smallmatrix}\right)(R)\right)$$

$$\begin{array}{ccc} \begin{pmatrix} S \\ S \end{pmatrix} & \Longrightarrow & \begin{pmatrix} R \\ R \end{pmatrix} \\ \updownarrow & & \updownarrow \\ \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{h^b} \\ \xrightarrow{h} \end{array} & \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \end{array}$$

$$\longmapsto$$

$$\begin{array}{ccc} \begin{pmatrix} S \\ S \end{pmatrix} & \Longrightarrow & \begin{pmatrix} S \\ S \end{pmatrix} \\ \updownarrow & & \updownarrow \\ \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{h^b} \\ \xrightarrow{h} \end{array} & \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \\ p^\# \updownarrow p & & q^\# \updownarrow q \\ \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \begin{array}{c} \xrightarrow{k^b} \\ \xrightarrow{k} \end{array} & \begin{pmatrix} D^- \\ D^+ \end{pmatrix} \end{array}$$

## Example: Moore machines

This data defines the **theory of Moore machines**:

$$\text{Moore} : \text{Lens}^{\top} \longrightarrow \text{Cat}$$

The **theory of non-deterministic Moore machines** is defined likewise:

$$\text{Moore}_{\mathcal{P}} : \text{Lens}_{\mathcal{P}}^{\top} \longrightarrow \text{Cat}$$

## Example: Moore machines

This data defines the **theory of Moore machines**:

$$\mathbf{Moore} : \mathbf{Lens}^{\top} \longrightarrow \mathbf{Cat}$$

The **theory of non-deterministic Moore machines** is defined likewise:

$$\mathbf{Moore}_{\mathcal{P}} : \mathbf{Lens}_{\mathcal{P}}^{\top} \longrightarrow \mathbf{Cat}$$

For the theory of 'dependent' Moore machines, we need an extra bit of data: a functor  $T : \mathcal{C} \rightarrow \int F$  that lifts a state space  $S : \mathcal{C}$  to a 'change space'  $TS$ . Archetypal example is the tangent bundle section  $T : \mathbf{Smooth} \rightarrow \int \mathbf{Sub}$ .

$$\mathbf{Moore}_{F,T} : \mathbf{Lens}_F^{\top} \longrightarrow \mathbf{Cat}$$

## Example: observational theory

Any observational theory of processes  $\mathbb{S}\text{pan}(\mathcal{C})$  supports a theory of observational systems.

In this theory, a system  $S$  over the interface  $I : \mathcal{C}$  is simply a state space  $S : \mathcal{C}$  together with a morphism  $\text{observe}_S : S \rightarrow I$ , projecting a state to its observable behaviour on the boundary.

Morphisms are commutative triangles.

$$\mathbb{O}\text{bs}_{\mathcal{C}}(I) = \left\{ \begin{array}{ccc} S & \xrightarrow{h} & R \\ \text{observe}_S \downarrow & & \downarrow \text{observe}_R \\ I & \equiv & I \end{array} \right\}$$

One can see maps  $S \rightarrow I$  as spans  $S \equiv S \rightarrow I$ , thereby fitting this example into a more general pattern of 'systems are processes with a special left boundary'

## Example: observational theory

Given a span  $I \xleftarrow{f_\ell} X \xrightarrow{f_r} J$ , we can reindex by pull-push (span composition):

$$\begin{array}{ccccc} S & \longleftarrow & f_\ell^* S & & \\ \text{observes} \downarrow & & \downarrow & \lrcorner & \\ I & \xleftarrow{f_\ell} & X & \xrightarrow{f_r} & J \end{array}$$

## Example: observational theory

Given a span  $I \xleftarrow{f_\ell} X \xrightarrow{f_r} J$ , we can reindex by pull-push (span composition):

$$\begin{array}{ccc} \begin{array}{c} S \\ \text{observes} \downarrow \\ I \end{array} & \xrightarrow{\quad} & \begin{array}{c} f_\ell^* S \\ \downarrow \text{observes} \\ X \\ \downarrow f_r \\ J \end{array} \\ \mathbb{O}\mathbf{b}\mathbf{s}_c(I) & \xrightarrow{\mathbb{O}\mathbf{b}\mathbf{s}_c(I \xleftarrow{f_\ell} X \xrightarrow{f_r} J)} & \mathbb{O}\mathbf{b}\mathbf{s}_c(J) \end{array}$$

## Example: observational theory

Given a map  $I \xrightarrow{h} K$ , we define a profunctor  $\mathbb{O}bs_C(h) : \mathbb{O}bs_C(I) \dashv\dashv \mathbb{O}bs_C(K)$ :

$$\mathbb{O}bs_C(I)^{op} \times \mathbb{O}bs_C(K) \xrightarrow{\mathbb{O}bs_C(h)} \mathbf{Set}$$

$$\begin{array}{ccc} \begin{array}{c} S \\ \text{observes}_S \downarrow \\ I \end{array} & \begin{array}{c} R \\ \text{observe}_R \downarrow \\ K \end{array} & \longmapsto \left\{ \begin{array}{ccc} S & \rightarrow & R \\ \text{observes}_S \downarrow & & \downarrow \text{observe}_R \\ J & \xrightarrow{h} & K \end{array} \right\} \end{array}$$

## Example: observational theory

Finally, we have a map on squares that sends

$$\begin{array}{ccc}
 I \xrightarrow{h} K & & \mathbb{O}bs_C(I) \xrightarrow{\mathbb{O}bs_C(f)} \mathbb{O}bs_C(J) \\
 f_\ell \uparrow & & \downarrow \mathbb{O}bs_C(h) \\
 X \xrightarrow{\sigma} Y & \mapsto & \xrightarrow{\mathbb{O}bs_C(\sigma)} \downarrow \mathbb{O}bs_C(k) \\
 f_r \downarrow & & \downarrow \\
 J \xrightarrow[k]{} L & & \mathbb{O}bs_C(K) \xrightarrow[\mathbb{O}bs_C(g)]{} \mathbb{O}bs_C(L)
 \end{array}$$

again by stacking:  $\mathbb{O}bs_C(h)(S, R) \xrightarrow{\mathbb{O}bs_C(\sigma)_{S,R}} \mathbb{O}bs_C(k)(\mathbb{O}bs_C(f)(S), \mathbb{O}bs_C(g)(R))$

$$\begin{array}{ccc}
 S \longrightarrow R & & S \longrightarrow R \\
 \text{observes} \downarrow & \text{observer}_R \downarrow & \downarrow \\
 I \xrightarrow{h} K & \mapsto & I \xrightarrow{h} K \\
 & & \uparrow \\
 & & X \xrightarrow{\sigma} Y \\
 & & \downarrow \\
 & & J \xrightarrow[k]{} L
 \end{array}$$

## Example: observational theory

Thus we have a system theory:

$$\mathbb{O}bs_{\mathcal{C}} : \mathbf{Span}(\mathcal{C})^{\top} \longrightarrow \mathbf{Cat}$$

In particular, we have

$$\mathbb{O}bs_{\mathbf{Set}} : \mathbf{Set}^{\top} \longrightarrow \mathbf{Cat}$$

which on objects is defined as  $I \mapsto \mathbf{Set}/I \cong \mathbf{Set}^I$ .

When we write  $\mathbb{O}bs$ , this is what we mean.

## The special role of the observational theory

Given  $\mathbb{S}\mathbf{y}\mathbf{s} : \mathbb{P}^\top \rightarrow \mathbb{C}\mathbf{a}\mathbf{t}$ , and chosen a system  $S : \mathbb{S}\mathbf{y}\mathbf{s}(I)$ , there is a triangle:

$$\begin{array}{ccc}
 \mathbb{P}^\top & \xrightarrow{\mathbb{P}^h(I, -)^\top} & \mathbf{S}\mathbf{e}\mathbf{t}^\top \\
 \mathbb{S}\mathbf{y}\mathbf{s} \downarrow & \xRightarrow{\mathbb{S}\mathbf{y}\mathbf{s}(S, -)} & \downarrow \mathbb{O}\mathbf{b}\mathbf{s} \\
 \mathbf{C}\mathbf{a}\mathbf{t} & \xlongequal{\quad} & \mathbf{C}\mathbf{a}\mathbf{t}
 \end{array}$$

where, for a given  $J : \mathbb{P}$ ,  $\mathbb{S}\mathbf{y}\mathbf{s}(S, -)_J : \mathbb{S}\mathbf{y}\mathbf{s}(J) \rightarrow \mathbb{O}\mathbf{b}\mathbf{s}(\mathbb{P}^h(I, J))$  is a functor equivalently given as

$$\begin{aligned}
 \mathbb{S}\mathbf{y}\mathbf{s}(J) \times \mathbb{P}^h(I, J) & \xrightarrow{\mathbb{S}\mathbf{y}\mathbf{s}(S, -)_J} \mathbf{S}\mathbf{e}\mathbf{t} \\
 (\mathbb{T}, \quad I \xrightarrow{h} J) & \mapsto \mathbb{S}\mathbf{y}\mathbf{s}(h)(S, \mathbb{T})
 \end{aligned}$$

This picks all simulations of  $S$  in  $\mathbb{T}$ , given a boundary map.

We call this **corepresentable behaviours of type  $S$** .

## Example: steady states behaviours

Let  $\text{fix} : \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 \\ 1 \end{pmatrix}$  be the trivial Moore machine (only one states, no input output).

Then  $\text{Moore}(\text{fix}, -)$  is the 'functor' of steady states: given another Moore machine  $T : \begin{pmatrix} I \\ O \end{pmatrix}$ , and a

chart  $\begin{pmatrix} i \\ o \end{pmatrix} : \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} I \\ O \end{pmatrix}$ , the set

$$\begin{aligned} \text{Moore}\left(\begin{pmatrix} i \\ o \end{pmatrix}\right)(\text{fix}, T) &= \left\{ t \in T \left| \begin{array}{ccc} \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \xrightarrow[\text{s}]{\pi_2 \circ \text{s}} & \begin{pmatrix} T \\ T \end{pmatrix} \\ || & & p^\# \uparrow \downarrow p \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \xrightarrow[o]{i} & \begin{pmatrix} I \\ O \end{pmatrix} \end{array} \right\} \\ &= \{t \in T \mid p(s) = o, p^\#(t, i) = t\} \end{aligned}$$

is the set of steady states for input  $i$  (and giving output  $o$ ).

## Example: fixpoint behaviours

Hence: **studying fixpoints of Moore machines amounts to studying**

$$(\text{view}, \text{Moore}(\text{fix}, -)) : \text{Moore} \longrightarrow \text{Obs}$$

In particular, we can automatically mint lots of compositional structure. For instance, we know for any

lens  $\left(\begin{smallmatrix} f^\sharp \\ f \end{smallmatrix}\right) : \left(\begin{smallmatrix} I \\ O \end{smallmatrix}\right) \rightleftharpoons \left(\begin{smallmatrix} J \\ Q \end{smallmatrix}\right)$  there is a natural map

$$\begin{array}{ccc} \text{Moore}\left(\begin{smallmatrix} I \\ O \end{smallmatrix}\right) & \xrightarrow{\text{Moore}(\text{fix}, -)_I} & \text{Set}^{O \times I} \\ \text{Moore}\left(\begin{smallmatrix} f^\sharp \\ f \end{smallmatrix}\right) \downarrow & & \downarrow f_* \\ \text{Moore}\left(\begin{smallmatrix} J \\ Q \end{smallmatrix}\right) & \xrightarrow{\text{Moore}(\text{fix}, -)_J} & \text{Set}^{Q \times J} \end{array}$$

$\text{Moore}\left(\text{fix}, \left(\begin{smallmatrix} f^\sharp \\ f \end{smallmatrix}\right)\right)$  (diagonal arrow)

sending fixpoints of  $S : \text{Moore}\left(\begin{smallmatrix} I \\ O \end{smallmatrix}\right)$  to fixpoints of its rewiring  $\text{Moore}\left(\begin{smallmatrix} f^\sharp \\ f \end{smallmatrix}\right)(S)$ .

**And one can prove in this case the map is iso!** This recovers [Spi15], and generalizes more (see [Mye22, Theorem 5.3.3.1]).

## Recap: categorical system theory

Categorical system theory studies system theories  $\mathbb{S}\text{ys} : \mathbb{P}^\top \rightarrow \mathbb{C}\text{at}$ , especially maps

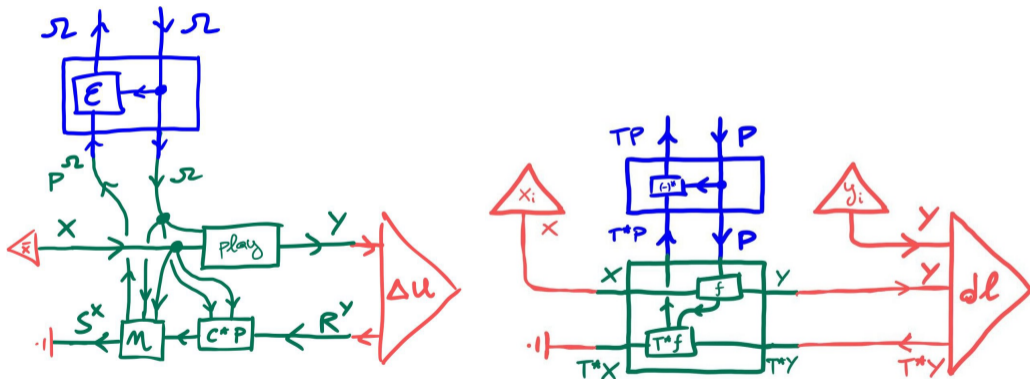
$$\begin{array}{ccc} \mathbb{P}^\top & \xrightarrow{B^\top} & \mathbb{S}\text{et}^\top \\ \mathbb{S}\text{ys} \downarrow & \xRightarrow{B^b} & \downarrow \text{Obs} \\ \mathbb{C}\text{at} & \xlongequal{\quad} & \mathbb{C}\text{at} \end{array}$$

Some of these are given to us automatically: they are the 'corepresentables' associated to any system  $S : \mathbb{S}\text{ys}$ .

# Categorical cybernetics

## Motivation

The systems of categorical system theory are varied and numerous, but they miss some interesting examples. In my work I mostly care about **games** and **learners**:



These (and others) are what I call **cybernetic systems**.

## Categorical cybernetics

The conceptual foundation of 'categorical cybernetics', as advocated in [Cap+21a] and in [Smi21] rests on two main pillars:

## Categorical cybernetics

The conceptual foundation of 'categorical cybernetics', as advocated in [Cap+21a] and in [Smi21] rests on two main pillars:

1. The fact cybernetic processes are *mereologically* peculiar in having a distinctive boundary between a 'controller system' and a 'controlled process';

## Categorical cybernetics

The conceptual foundation of 'categorical cybernetics', as advocated in [Cap+21a] and in [Smi21] rests on two main pillars:

1. The fact cybernetic processes are *mereologically* peculiar in having a distinctive boundary between a 'controller system' and a 'controlled process';
2. The fact cybernetic processes tend to organize in **bicategorical structures**, where the first dimension ignores the mereological distinction between controller and controllee, and the second dimension distinguishes the controller system—so that higher dimensions encode deeper control hierarchies;

## Categorical cybernetics

My favourite way to create bicategories of cybernetic processes is by using the **Para** construction. A process in  $\mathbf{Para}(\mathcal{C})$ , for  $\mathcal{C}$  monoidal, has the same objects as  $\mathcal{C}$  but morphisms have one extra input, the **parameter**

$$\mathbf{Para}(\mathcal{C})(X, Y) = \sum_{P:\mathcal{C}} \mathcal{C}(P \otimes X, Y)$$

Then 2-morphisms are given by reparametrisations, i.e. morphisms involving only the parameter:

$$\begin{array}{ccc} X & & X \\ \downarrow f & \xrightarrow{\phi} & \downarrow g \\ Y & & Y \end{array} := \begin{array}{ccc} Q \otimes X & \xrightarrow{\phi \otimes X} & P \otimes X \\ \downarrow (Q, f) & & \downarrow (P, g) \\ Y & & Y \end{array}$$

## Categorical cybernetics

With little effort, we can directly promote  $\mathbf{Para}(\mathcal{C})$  to be a double category  $\mathbb{P}\mathbf{ara}(\mathcal{C})$ :

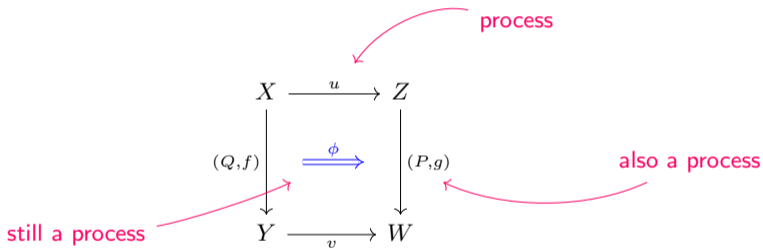
$$\begin{array}{ccc} X & \xrightarrow{u} & Z \\ \downarrow (Q,f) & \xRightarrow{\phi} & \downarrow (P,g) \\ Y & \xrightarrow{v} & W \end{array} := \begin{array}{ccc} Q \otimes X & \xrightarrow{\phi \otimes u} & P \otimes X \\ \downarrow f & & \downarrow g \\ Y & \xrightarrow{v} & W \end{array}$$

And not just that, but a **company** (a double category where all vertical arrows have companions), since every non-parametric morphism is a trivially parametric morphism too.

This structure brings together other variations on the **Para** construction, like Smithe's 'external **Para**' [Smi21], **Span**, etc.

## Categorical cybernetics

Hence a company could be a good candidate for an objective definition of cybernetic process theory, except we lack maps of interfaces and of processes which are not themselves processes.



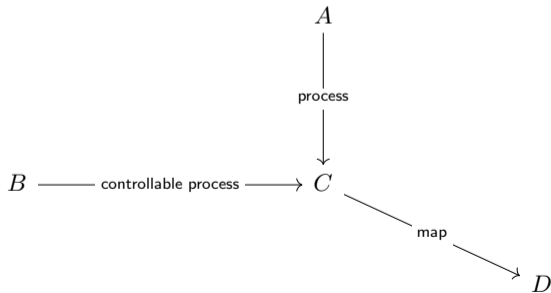
**Thus we are forced to go 3D!**

## Cybernetic process theories

### Definition

A **cybernetic process theory** is a triple category with attitude where

1. objects are **boundaries/interfaces**,
2. transversal 2-cells are **maps of boundaries**,
3. vertical 1-cells are **processes**,
4. horizontal 1-cells are **controllable processes**,

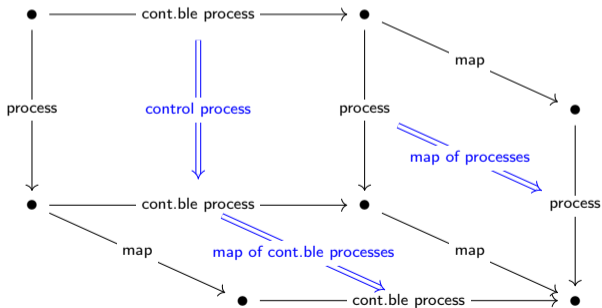


# Cybernetic process theories

## Definition

A **cybernetic process theory** is a triple category with attitude where

1. basal 2-cells are **control processes** (and form a *company*),
2. vertical 2-cells are **maps of processes**,
3. horizontal 2-cells are **maps of controllable processes**

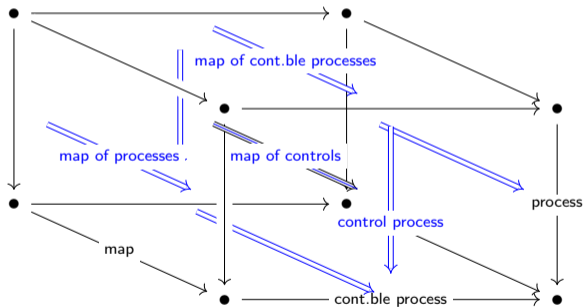


# Cybernetic process theories

## Definition

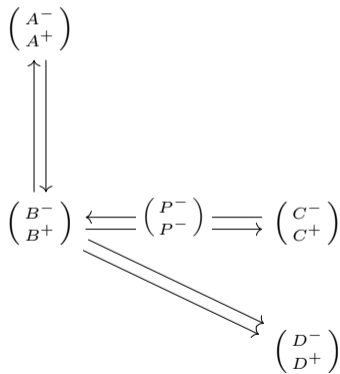
A **cybernetic process theory** is a triple category with attitude where

1. cubes are **maps of control processes**



## Example: parametric lenses

One can promote the process theory of lenses to a cybernetic theory  $\mathbb{P}\text{ara}(\mathbb{L}\text{ens})^1$ , with parametric lenses in the role of controllable processes:

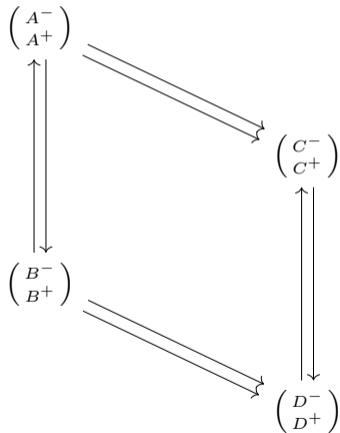


---

<sup>1</sup>This notation is actually sound: there's a Para construction that yields this triple category

## Example: parametric lenses

Vertical 2-cells are the same kind we encountered before ('commutative squares'):



## Example: parametric lenses

Horizontal 2-cells are still of the same kind, but there's an extra chart going between the parameters of the parametric lenses:

$$\begin{array}{ccccc} \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \begin{array}{c} \longleftarrow \\ \longrightarrow \end{array} & \begin{pmatrix} P^+ \\ P^- \end{pmatrix} & \begin{array}{c} \longrightarrow \\ \longleftarrow \end{array} & \begin{pmatrix} E^- \\ E^+ \end{pmatrix} \\ & \searrow & \begin{array}{c} \text{blue} \\ \text{double} \\ \text{arrow} \end{array} & \begin{array}{c} \text{blue} \\ \text{double} \\ \text{arrow} \end{array} & \searrow \\ & & \begin{pmatrix} h^b \\ h \end{pmatrix} & \begin{array}{c} \text{blue} \\ \text{double} \\ \text{arrow} \end{array} & \\ & & \begin{pmatrix} D^- \\ D^+ \end{pmatrix} & \begin{array}{c} \longleftarrow \\ \longrightarrow \end{array} & \begin{pmatrix} Q^- \\ Q^+ \end{pmatrix} & \begin{array}{c} \longrightarrow \\ \longleftarrow \end{array} & \begin{pmatrix} F^- \\ F^+ \end{pmatrix} \end{array}$$

## Example: parametric lenses

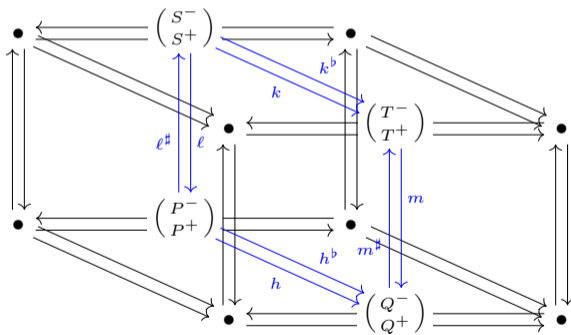
Basic 2-cells are commutative squares of parametric lenses:

$$\begin{array}{ccccc} \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \longleftarrow & \begin{pmatrix} S^- \\ S^+ \end{pmatrix} & \longrightarrow & \begin{pmatrix} G^- \\ G^+ \end{pmatrix} \\ \updownarrow & & \updownarrow & & \updownarrow \\ \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \longleftarrow & \begin{pmatrix} P^- \\ P^+ \end{pmatrix} & \longrightarrow & \begin{pmatrix} E^- \\ E^+ \end{pmatrix} \end{array}$$

*ℓ<sup>#</sup>*   *ℓ*

## Example: parametric lenses

A cube is an arrangement of faces such that the blue parts form a square in  $\mathbb{L}ens$ :



## Aside: Para(Lens) and Org

In [SS22], Spivak and Shapiro define a double category  $\mathbb{O}rg$  where

1. objects are *polynomial functors*, i.e. functors of the form  $p = \sum_{i:p(1)} y^{p[i]}$
2. loose arrows  $(S, \phi) : p \dashrightarrow q$  are *polynomial coalgebras*, i.e. coalgebras of the form

$$S : \mathbf{Set}, \quad \phi : S \longrightarrow [p, q](S)$$

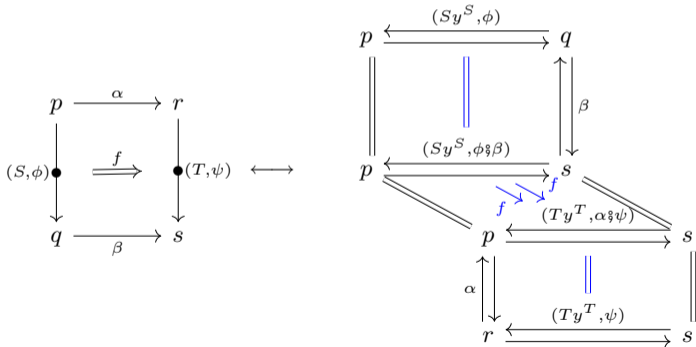
where  $[-, -]$  is the closed structure associated to the Hancock product,

3. tight arrows  $h : p \rightarrow r$  are morphisms of polynomial functors,
4. squares are given by maps between the carriers of the coalgebras, plus a commutativity condition:

$$\begin{array}{ccc}
 p & \xrightarrow{\alpha} & r \\
 \downarrow & & \downarrow \\
 (S, \phi) \bullet & \xrightarrow{f} & \bullet (T, \psi) \\
 \downarrow & & \downarrow \\
 q & \xrightarrow{\beta} & s
 \end{array}
 \quad := \quad
 \begin{array}{ccc}
 S & \xrightarrow{f} & T \\
 \phi \downarrow & & \downarrow \psi \\
 [p, q](S) & & [r, s](T) \\
 [p, \beta](S) \downarrow & & \downarrow [\alpha, s](T) \\
 [p, s](S) & \xrightarrow{[p, s](f)} & [p, s](T)
 \end{array}$$

## Example: Para(Lens) and Org

Recalling that  $\mathbf{Poly} = \mathbf{Lens}_{\mathbf{Set}/-}$ , and that polynomial coalgebras can equivalently be given as parametric maps  $Sy^S \otimes p \rightarrow q$ , and that coalgebra maps between them become *charts*, we see that  $\mathbf{Org}$  embeds in  $\mathbf{Para}(\mathbf{Lens}_{\mathbf{Set}/-})$  'diagonally':



## Example: Para(Lens) and Org

Recalling that  $\mathbf{Poly} = \mathbf{Lens}_{\mathbf{Set}/-}$ , and that polynomial coalgebras can equivalently be given as parametric maps  $Sy^S \otimes p \rightarrow q$ , and that coalgebra maps between them become *charts*, we see that  $\mathbf{Org}$  embeds in  $\mathbf{Para}(\mathbf{Lens}_{\mathbf{Set}/-})$  'diagonally':

$$\begin{array}{ccc}
 p & \xlongequal{\quad} & p \\
 (S, \phi) \downarrow & & \downarrow \alpha \\
 q & \xrightarrow{\quad f \quad} & r \\
 \beta \downarrow & & \downarrow (T, \psi) \\
 s & \xlongequal{\quad} & s
 \end{array}
 \longleftrightarrow
 \begin{array}{ccc}
 & (Sy^S, \phi \natural \beta) & \\
 p & \xrightarrow{\quad} & s \\
 & \searrow & \swarrow \\
 & p & s \\
 & \xrightarrow{\quad} & \\
 & (Ty^T, \alpha \natural \psi) & 
 \end{array}$$

## Example: Nash equilibria

Let's see an example of cube in  $\mathbb{P}\text{ara}(\mathbb{L}\text{ens}_P)$ .

A classical game can be represented by a basal 2-cell in this category:

$$\begin{array}{ccccc} \left( \begin{array}{c} \mathbb{R}^{N \times X} \\ X \end{array} \right) & \longleftarrow & \left( \begin{array}{c} \Omega \\ \Omega \end{array} \right) & \longrightarrow & \left( \begin{array}{c} \mathbb{R}^{N \times Y} \\ Y \end{array} \right) \\ \parallel & & \uparrow & & \parallel \\ \left( \begin{array}{c} \mathbb{R}^{N \times X} \\ X \end{array} \right) & \longleftarrow & \left( \begin{array}{c} \mathbb{R}^{N \times \Omega} \\ \Omega \end{array} \right) & \longrightarrow & \left( \begin{array}{c} \mathbb{R}^{N \times Y} \\ Y \end{array} \right) \end{array}$$

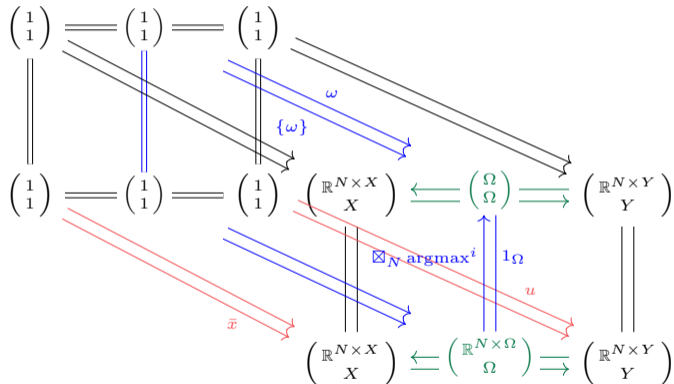
$\boxtimes_N \text{argmax}^i$        $1_\Omega$

Here we have  $N$  players, each maximizing their payoff independently, playing in an arena with initial states  $X$  and final outcomes  $Y$ . Their strategy profiles are  $\Omega = \Omega_1 \times \cdots \times \Omega_N$ .

I'm assuming this parametric lens has been built using the principles of [Cap22].

## Example: Nash equilibria

Now if we contemplate a cube whose back face is the 'trivial system' fix we get the following:



Here  $u : Y \rightarrow \mathbb{R}^N$  is a payoff function,  $\bar{x}$  an initial state, and  $\bar{\omega}$  a strategy profile.

## Example: Nash equilibria

Assuming the parametric lens at the basis has form  $\text{play} : X \times \Omega \rightarrow Y$ , concretely the commutativity condition on the cube imposes

$$\bar{\omega} \in \boxtimes_{i \in N} \operatorname{argmax}(\lambda \omega_i . u_i(\text{play}(\bar{x}, \bar{\omega}_1, \dots, \omega_i, \dots, \bar{\omega}_N)))$$

and by definition of  $\boxtimes$ , we get

$$\forall i \in N \bar{\omega}_i \in \operatorname{argmax}^i(\lambda \omega_i . u_i(\text{play}(\bar{x}, \bar{\omega}_1, \dots, \omega_i, \dots, \bar{\omega}_N)))$$

which is the definition of **Nash equilibrium**.

## Cybernetic system theory

We would like to recover the compositional formulae for Nash equilibria as functoriality of some corepresentable behaviour  $\mathbb{G}\mathbf{ames}(\text{fix}, -)$ , whatever this means.

## Cybernetic system theory

We would like to recover the compositional formulae for Nash equilibria as functoriality of some corepresentable behaviour  $\mathbb{G}\text{ames}(\text{fix}, -)$ , whatever this means.

This would require to come up with the same ingredients we've used for open dynamical systems:

1. A notion of **cybernetic system theory** over a given cybernetic process theory
2. A cybernetic theory of observational processes and systems
3. A notion of **behaviour functor** between the two

## Cybernetic system theory

We would like to recover the compositional formulae for Nash equilibria as functoriality of some corepresentable behaviour  $\mathbb{G}\text{ames}(\text{fix}, -)$ , whatever this means.

This would require to come up with the same ingredients we've used for open dynamical systems:

1. A notion of **cybernetic system theory** over a given cybernetic process theory
2. A cybernetic theory of observational processes and systems
3. A notion of **behaviour functor** between the two

**At the moment, I've not been able to do this yet!**

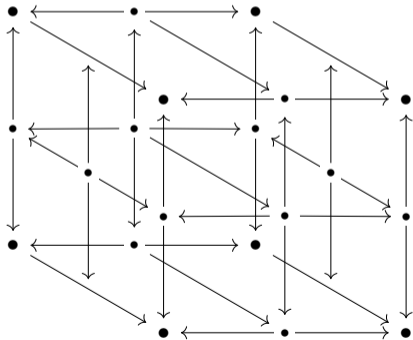
## Towards cybernetic system theory

But here's some insight...

## Towards cybernetic system theory

But here's some insight...

1. There is an obvious **theory of controllable observational processes** given by the triple category  $\mathbb{S}\text{pan}(\text{Set})$  of *spans of spans of sets*. The reason why this is an obvious choice is transversal hom-functors of triple categories land in  $\mathbb{S}\text{pan}(\text{Set})$  [GP17].



## Towards cybernetic system theory

But here's some insight...

2. A cybernetic system theory should arguably be talking about systems living over any given controllable process, i.e. be **control systems**. This suggests that if we ignore horizontal composition, so that controllable processes become *objects* (and hor. (resp. ver.) 2-cells hor. (resp. ver.) 1-cells, and cubes squares), we can look at functors very similar to the one we use in categorical system theory.

$$\begin{array}{ccc} \mathbf{Cyb}_1 & \longrightarrow & \mathbf{Cat} \\ I \xrightarrow{P} J & \mapsto & \mathbf{ContProc}(I \xrightarrow{P} J) \end{array}$$

e.g. taking a parametric lens  $\left( \begin{pmatrix} P^- \\ P^+ \end{pmatrix}, \begin{pmatrix} p^\# \\ p \end{pmatrix} \right) : \begin{pmatrix} I^- \\ I^+ \end{pmatrix} \rightleftarrows \begin{pmatrix} J^- \\ J^+ \end{pmatrix}$  to the category of reparametrisations with codomain  $\begin{pmatrix} TS \\ S \end{pmatrix}$ .

Then there is an operation:

$$\mathbf{ContProc}(I \xrightarrow{P} J) \times \mathbf{ContProc}(J \xrightarrow{Q} K) \longrightarrow \mathbf{ContProc}(I \xrightarrow{P \circ Q} J)$$

## Towards cybernetic system theory

But here's some insight...






3. Accordingly, if we take the 'vertical process theory' (the process theory given by the vertical 2-cells) of a given cybernetic process theory, and we look at systems for this, these should 'act on the left' of a given cybernetic system theory. This would amount to attach a stateful boundary to the left of a given controllable process.

$$\mathbf{ContSys}(I) \times \mathbf{ContProc}(I \xrightarrow{P} J) \longrightarrow \mathbf{ContSys}(J)$$


**Thanks for your attention!**

**Questions?**






## References |

-  J. C. Baez and K. Courser, “Structured cospans”, *Theory and Applications of Categories*, vol. 35, no. 48, pp. 1771–1822, 2020.
-  M. Capucci, “Parametrised categories and categories by proxy”, Slides at <https://matteocapucci.files.wordpress.com/2021/09/main-annotated.pdf>, 2021.
-  —, “Diegetic representation of feedback in open games”, *arXiv preprint arXiv:2206.12338*, 2022.
-  M. Capucci, B. Gavranović, J. Hedges, and E. F. Rischel, “Towards foundations of categorical cybernetics”, in *Proceedings of Applied Category Theory 2021*, EPTCS, 2021. [Online]. Available: <https://arxiv.org/abs/2105.06332>.
-  M. Capucci, N. Ghani, J. Ledent, and F. N. Forsberg, “Translating extensive form games to open games with agency”, in *Proceedings of Applied Category Theory 2021*, EPTCS, 2021. [Online]. Available: <https://arxiv.org/abs/2105.06763>.


## References II

-  G. S. H. Cruttwell, B. Gavranovic, N. Ghani, P. W. Wilson, and F. Zanasi, “Categorical foundations of gradient-based learning”, *CoRR*, vol. abs/2103.01931, 2021. arXiv: 2103.01931. [Online]. Available: <https://arxiv.org/abs/2103.01931>.
-  J. Culbertson, P. Gustafson, D. E. Koditschek, and P. F. Stiller, “Formal composition of hybrid systems”, *Theory and Applications of Categories*, vol. 35, no. 45, pp. 1634–1682, 2020.
-  D. Fudenberg and D. K. Levine, *The theory of learning in games*. MIT press, 1998, vol. 2.
-  M. Grandis, *Higher Dimensional Categories: From Double to Multiple Categories*. World Scientific, 2019.
-  M. Grandis and R. Paré, “Intercategories: A framework for three-dimensional category theory”, *Journal of Pure and Applied Algebra*, vol. 221, no. 5, pp. 999–1054, 2017.
-  E. Lerman, “Networks of open systems”, *Journal of Geometry and Physics*, vol. 130, pp. 81–112, 2018.

## References III

-  K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Morgan & Claypool, 2008.
-  D. J. Myers, “Double categories of open dynamical systems”, 2020, [Online]. Available: <https://arxiv.org/abs/2005.05956>.
-  —, *Categorical Systems Theory*. 2022, (Work in progress). [Online]. Available: <http://davidjaz.com/Papers/DynamicalBook.pdf> (visited on 05/14/2022).
-  J. Nash, “Non-cooperative games”, *Annals of mathematics*, pp. 286–295, 1951.
-  B. Shapiro and D. I. Spivak, “Dynamic categories, dynamic operads: From deep learning to prediction markets”, *arXiv preprint arXiv:2205.03906*, 2022.
-  T. S. C. Smithe, “Compositional active inference i: Bayesian lenses. statistical games”, 2021, [Online]. Available: <https://arxiv.org/abs/2109.04461>.
-  —, “Open dynamical systems as coalgebras for polynomial functors, with application to predictive processing”, *arXiv preprint arXiv:2206.03868*, 2022.

## References IV

-  D. I. Spivak, “The steady states of coupled dynamical systems compose according to matrix arithmetic”, *arXiv preprint arXiv:1512.00802*, 2015.