From categorical systems theory to categorical cybernetics

Matteo Capucci

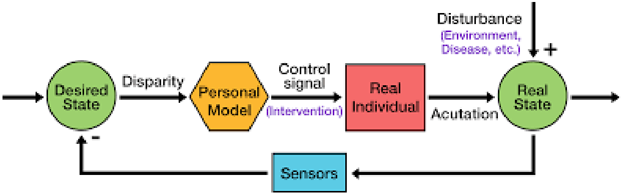
MSP group, University of Strathclyde

Virtual Double Categories Workshop

December 2nd, 2022

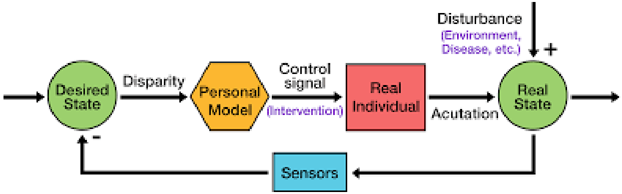
Cybernetic systems are

systems with control mechanisms and interactive feedback



Cybernetic systems are

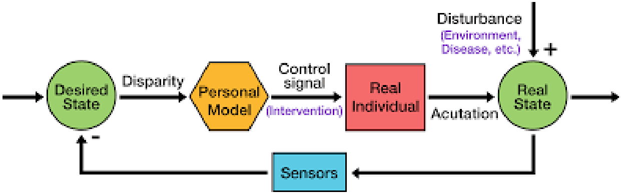
systems with control mechanisms and interactive feedback



control theory

Cybernetic systems are

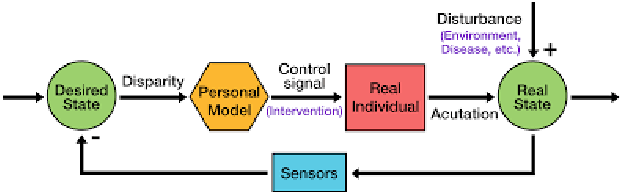
systems with control mechanisms and interactive feedback



control theory, **game theory**

Cybernetic systems are

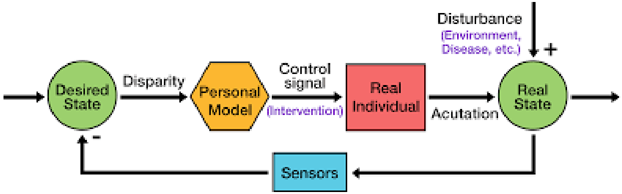
systems with control mechanisms and interactive feedback



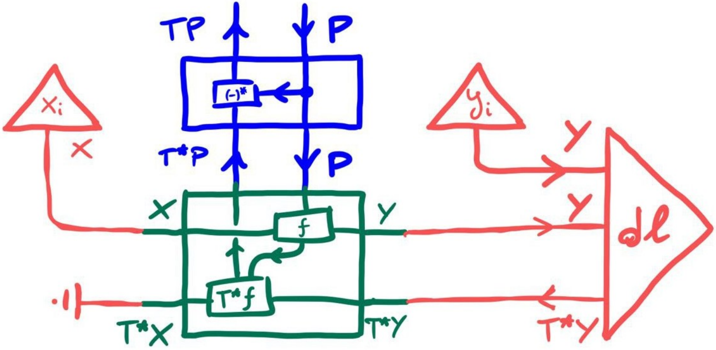
control theory, **game theory,** **learning theory**

Cybernetic systems are

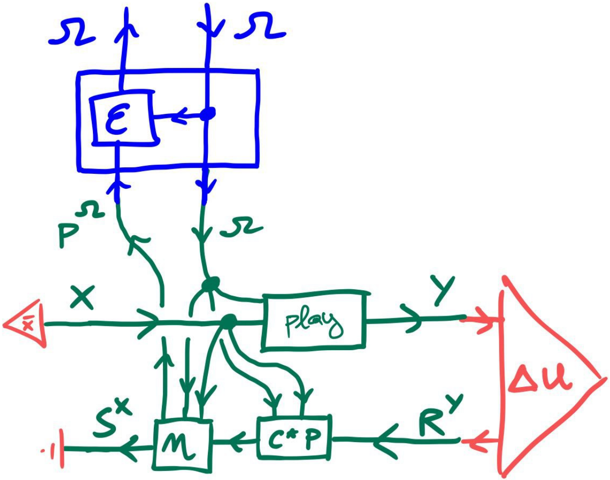
systems with control mechanisms and interactive feedback



control theory, game theory, learning theory, etc...



A model trained by gradient-descent
(Fong, D. Spivak, and Tuyéras 2019; Cruttwell, Gavranović, Ghani, Wilson, and Zanasi 2021; Capucci 2022)



A strategic game

(Ghani, Hedges, Winschel, and Zahn 2018; Capucci, Gavranović, Hedges, and Rischel 2021; Capucci 2022)

But...

What does it mean to ‘study cybernetic systems’?

But...

What does it mean to 'study cybernetic systems'?

Ontology

What 'is' a cybernetic system?

What is it made of?

How does it interact with other cybernetic systems?

What is its interface? How do we describe it?

But...

What does it mean to 'study cybernetic systems'?

Ontology

What 'is' a cybernetic system?

What is it made of?

How does it interact with other cybernetic systems?

What is its interface? How do we describe it?

Phenomenology

What does a cybernetic system 'do'?

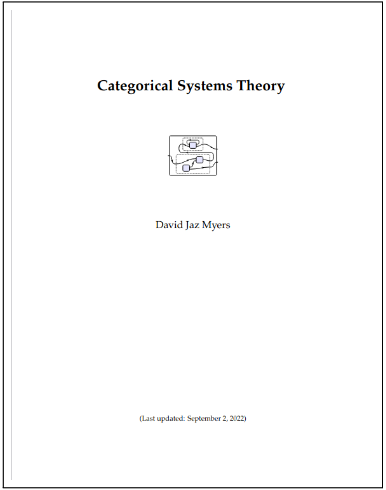
What can it be observed about it?

How do observations of different parts relate to observations on the whole?

When are two cybernetic systems observationally interchangeable?

Categorical Systems Theory

A principled mathematical framework for the ontology and phenomenology of systems is given by **categorical systems theory** (CST)


Categorical Systems Theory

David Jaz Myers

(Last updated: September 2, 2022)

Myers 2021

Categorical Systems Theory

CST is a principled and widely applicable paradigm for organizing a ‘theory’ of systems/processes.

It encompasses any example I found in the literature so far:

Categorical Systems Theory

CST is a principled and widely applicable paradigm for organizing a ‘theory’ of systems/processes.

It encompasses any example I found in the literature so far:

1. **Algebras for operads of string diagrams** (D. I. Spivak 2013; Vagner, D. I. Spivak, and Lerman 2014; Libkind, Baas, Patterson, and Fairbanks 2021; Shapiro and D. I. Spivak 2022)

Categorical Systems Theory

CST is a principled and widely applicable paradigm for organizing a 'theory' of systems/processes.

It encompasses any example I found in the literature so far:

1. **Algebras for operads of string diagrams** (D. I. Spivak 2013; Vagner, D. I. Spivak, and Lerman 2014; Libkind, Baas, Patterson, and Fairbanks 2021; Shapiro and D. I. Spivak 2022)
2. **Bicategories of transition systems** (Katis, Sabadini, and Walters 1997a; Katis, Sabadini, and Walters 1997b; Katis, Sabadini, and Walters 2002; Gianola, Kasangian, and Sabadini 2017; Di Lavore, Gianola, Román, Sabadini, and Sobociński 2021)

Categorical Systems Theory

CST is a principled and widely applicable paradigm for organizing a 'theory' of systems/processes.
It encompasses any example I found in the literature so far:

1. **Algebras for operads of string diagrams** (D. I. Spivak 2013; Vagner, D. I. Spivak, and Lerman 2014; Libkind, Baas, Patterson, and Fairbanks 2021; Shapiro and D. I. Spivak 2022)
2. **Bicategories of transition systems** (Katis, Sabadini, and Walters 1997a; Katis, Sabadini, and Walters 1997b; Katis, Sabadini, and Walters 2002; Gianola, Kasangian, and Sabadini 2017; Di Lavore, Gianola, Román, Sabadini, and Sobociński 2021)
3. **Double categories of structured cospans** (Fiadeiro and Schmitt 2007; Fong 2015; Baez and Courser 2020; Baez, Courser, and Vasilakopoulou 2022; Baez and Master 2020)

Categorical Systems Theory

CST is a principled and widely applicable paradigm for organizing a 'theory' of systems/processes.
It encompasses any example I found in the literature so far:

1. **Algebras for operads of string diagrams** (D. I. Spivak 2013; Vagner, D. I. Spivak, and Lerman 2014; Libkind, Baas, Patterson, and Fairbanks 2021; Shapiro and D. I. Spivak 2022)
2. **Bicategories of transition systems** (Katis, Sabadini, and Walters 1997a; Katis, Sabadini, and Walters 1997b; Katis, Sabadini, and Walters 2002; Gianola, Kasangian, and Sabadini 2017; Di Lavore, Gianola, Román, Sabadini, and Sobociński 2021)
3. **Double categories of structured cospans** (Fiadeiro and Schmitt 2007; Fong 2015; Baez and Courser 2020; Baez, Courser, and Vasilakopoulou 2022; Baez and Master 2020)
4. **Symmetric monoidal categories** (Abramsky and Coecke 2004; Coecke and Paquette 2010; Coecke and Kissinger 2018)

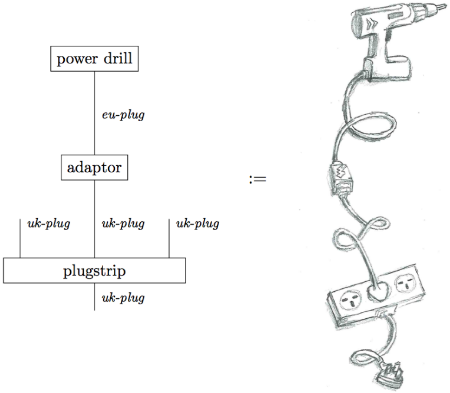
Categorical Systems Theory

In fact, CST is inspired by characteristics shared across them all:

Categorical Systems Theory

In fact, CST is inspired by characteristics shared across them all:

1. **Processes compose like morphisms of a category**



From (Coecke and Kissinger 2018)

Categorical Systems Theory

In fact, CST is inspired by characteristics shared across them all:

1. **Processes compose like morphisms of a category**
2. **Behaviours are relations/spans/matrices**, and are (lax) functorially determined:

If $(U, \alpha) : X \to Y$ is a circuit, an element of U is called a *state* of (U, α), and an element of X or Y is referred to as an *input* or an *output* for (U, α) respectively.

Definition 3. The following data define the homomorphism

$$\text{equilibrium} : \mathbf{Circ} \to \mathbf{Span}.$$

(a) If X is an object of **Circ**, $\text{equilibrium}(X) = X$.
(b) If $(U, \alpha) : X \to Y$ is a circuit, let

$$W = \{(x, u) \in XU \mid p_U \cdot \alpha(x, u) = u\},$$

$$p : W \to X : (x, u) \mapsto x,$$

and

$$q : W \to Y : (x, u) \mapsto p_Y \cdot \alpha(x, u).$$

From (Katis, Sabadini, and Walters 1997a)

Categorical Systems Theory

In fact, CST is inspired by characteristics shared across them all:

1. **Processes compose like morphisms of a category**
2. **Behaviours are relations/spans/matrices**, and are (lax) functorially determined:

In Theorem 23, we show that the map sending P to $\blacksquare P$ extends to a lax double functor

$$\blacksquare \colon \mathbb{O}\mathbf{pen}(\mathsf{Petri}) \to \mathbb{R}\mathbf{el}.$$

From (Baez and Master 2020)

Categorical Systems Theory

In fact, CST is inspired by characteristics shared across them all:

1. **Processes compose like morphisms of a category**
2. **Behaviours are relations/spans/matrices**, and are (lax) functorially determined:

Definition 2.28. Let $F = (S, f^{\mathrm{rdt}}, f^{\mathrm{upd}})$ be an (A, B)-discrete system. For $a \in A$ and $b \in B$, recall the set of (a, b)-steady states from Definition 2.4 and its count

$$\mathrm{Stst}(F)_{a,b} = \#\{s \in S \mid f^{\mathrm{rdt}}(s) = b \text{ and } f^{\mathrm{upd}}(a, s) = s\}$$

We can consider this as a matrix $\mathrm{Stst}(F) \in \mathsf{Mat}(A, B)$, which we call the *steady state matrix of* F.

From (D. I. Spivak 2015)

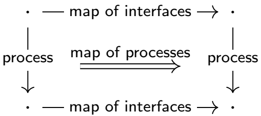
Categorical Systems Theory

However, CST adds two important principles:

Categorical Systems Theory

However, CST adds two important principles:

1. **Processes organize in double categories**.



This explains why behaviours land in $\mathbf{Set}\text{-}\mathbf{Mat}/\mathbb{S}\mathbf{pan}(\mathbf{Set})$: behaviours are often horizontal corepresentables!

Categorical Systems Theory

However, CST adds two important principles:

1. **Processes organize in double categories**.

   This explains why behaviours land in $\mathbf{Set}\text{-}\mathbf{Mat}/\mathbb{S}\mathbf{pan}(\mathbf{Set})$: behaviours are often horizontal corepresentables!

2. **There is a distinction between processes and systems**.

   Processes mediate interaction between systems, by acting on their interfaces:

$$\mathbf{Sys} : \mathbb{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

Categorical Systems Theory

However, CST adds two important principles:

1. **Processes organize in double categories.**
   This explains why behaviours land in $\mathbf{Set}\text{-}\mathbf{Mat}/\mathbb{S}\mathbf{pan}(\mathbf{Set})$: behaviours are often horizontal corepresentables!
2. **There is a distinction between processes and systems.**
   Processes mediate interaction between systems, by acting on their interfaces:

$$\mathbf{Sys} : \mathbb{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

Example

Dynamical systems (Rosen 1978) are endomorphisms $T : S \to S$ on some objects of states, hopefully equipped with an observable $X : S \to O$. The notion of process between these systems is functions $O \to O'$ that remap observations. Very different!

Plan of the talk

The talk is divided in two parts:

1. An introduction to **categorical systems theory**:

Plan of the talk

The talk is divided in two parts:

1. An introduction to **categorical systems theory**:
    1. Theories of processes and examples

Plan of the talk

The talk is divided in two parts:

1. An introduction to **categorical systems theory**:
   1. Theories of processes and examples
   2. Theories of systems and examples

Plan of the talk

The talk is divided in two parts:

1. An introduction to **categorical systems theory**:
    1. Theories of processes and examples
    2. Theories of systems and examples
    3. Functorial behaviours and examples

Plan of the talk

The talk is divided in two parts:

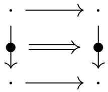
1. An introduction to **categorical systems theory**:
    1. Theories of processes and examples
    2. Theories of systems and examples
    3. Functorial behaviours and examples

    All in double dimension

2. A glimpse of **categorical cybernetic systems theory**,
   and why we need the triple dimension

Notation & conventions

1. Double categories are **weak** by default, (double) functors are **lax** by default
2. For the rest I mostly follow Grandis (2019)
3. ‘Loose’ arrows are vertical (denoted: $\mathrel{-\!\!\bullet\!\!\to}$), ‘tight’ arrows are horizontal (denoted: $\to$)



4. $\mathbb{S}\mathbf{et}$ denotes the **double category of spans** in $\mathbf{Set}$,
5. $\mathbb{C}\mathbf{at}$ denotes the **double category of** categories, functors, **profunctors** and natural transformations thereof.

I diverged from Myers (2021) on some notational and terminological choices.

Categorical systems theory

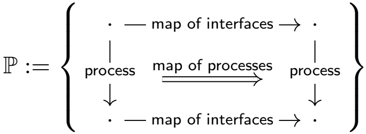
Motivation

Categorical systems theory is a conceptually simple, if mathematically sophisticated, framework:

Motivation

Categorical systems theory is a conceptually simple, if mathematically sophisticated, framework:

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



Motivation

Categorical systems theory is a conceptually simple, if mathematically sophisticated, framework:

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

$$\mathbb{P} := \left\{ \begin{array}{ccc} \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \\ \text{process}\big\downarrow & \overset{\text{map of processes}}{\Longrightarrow} & \big\downarrow\text{process} \\ \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \end{array} \right\}$$

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

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

Motivation

Categorical systems theory is a conceptually simple, if mathematically sophisticated, framework:

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

$$\mathbb{P} := \left\{ \begin{array}{ccc} \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \\ \Big\downarrow{\scriptstyle \text{process}} & \overset{\text{map of processes}}{\Longrightarrow} & \Big\downarrow{\scriptstyle \text{process}} \\ \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \end{array} \right\}$$

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

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

3. Finally, **behaviour is studied by describing maps into the span-ish ‘observational theory’**:

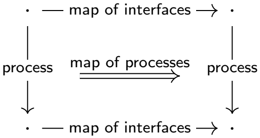
$$\begin{array}{ccc} \mathbb{P}^{\top} & \xrightarrow{B^{\top}} & \mathbb{S}\mathbf{et}^{\top} \\ \Big\downarrow{\scriptstyle \mathbf{Sys}} & \overset{B^{\flat}}{\Longrightarrow} & \Big\downarrow{\scriptstyle \mathbf{Obs}} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

Process theories

Definition

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

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

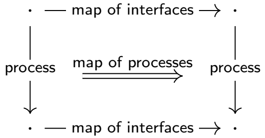


Process theories

Definition

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

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



Adding structure to the double category refines the kind of processes we are talking about,

e.g.: **concurrent process theories** have a monoidal product with the attitude of spatial juxtaposition of processes,

Example: lenses

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

Example: lenses

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

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

Example: lenses

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

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

$$\begin{pmatrix} A^- \\ A^+ \end{pmatrix} \underset{p}{\overset{p^\sharp}{\leftrightarrows}} \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

   where

$$p : A^+ \to B^+, \qquad p^\sharp : A^+ \times B^- \to A^-$$

Example: lenses

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

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

$$\begin{pmatrix} A^- \\ A^+ \end{pmatrix} \underset{p}{\overset{p^\sharp}{\leftrightarrows}} \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

   where

$$p : A^+ \to B^+, \qquad p^\sharp : A^+ \times B^- \to A^-$$

3. maps of boundaries are charts:

$$\begin{pmatrix} A^- \\ A^+ \end{pmatrix} \underset{h}{\overset{h^\flat}{\rightrightarrows}} \begin{pmatrix} C^- \\ C^+ \end{pmatrix}$$

   where

$$h : A^+ \to C^+, \qquad h^\flat : A^+ \times A^- \to C^-$$

Example: lenses

3. maps of processes are 'commutative squares':

$$\begin{array}{ccc} \begin{pmatrix} A^- \\ A^+ \end{pmatrix} & \overset{h^\flat}{\underset{h}{\rightleftarrows}} & \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \\ {\scriptstyle p^\sharp}\uparrow\downarrow{\scriptstyle p} & & {\scriptstyle q^\sharp}\uparrow\downarrow{\scriptstyle q} \\ \begin{pmatrix} B^- \\ B^+ \end{pmatrix} & \overset{k^\flat}{\underset{k}{\rightleftarrows}} & \begin{pmatrix} D^- \\ D^+ \end{pmatrix} \end{array}$$

meaning for every $a^+ \in A^+$ and $b^- \in B^-$ we have

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

Example: observational theories

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

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

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

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

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.

More generally, systems might be very different from the processes we use to plug them to!

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.
More generally, systems might be very different from the processes we use to plug them to!

Example

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

$$\begin{pmatrix} \text{update} \\ \text{observe} \end{pmatrix} : \begin{pmatrix} S \\ S \end{pmatrix} \leftrightarrows \begin{pmatrix} I \\ O \end{pmatrix}$$

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.
More generally, systems might be very different from the processes we use to plug them to!

Example

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

$$\begin{pmatrix} \text{update} \\ \text{observe} \end{pmatrix} : \begin{pmatrix} S \\ S \end{pmatrix} \leftrightarrows \begin{pmatrix} I \\ O \end{pmatrix}$$

This is an interactive finite state machine.

Example

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

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

Systems theories

Crucially, **systems** are

1. **Extendable** by processes going out of their interfaces,
2. **Comparable** with other systems given a way to compare their interfaces and states

Systems theories

Crucially, **systems** are

1. **Extendable** by processes going out of their interfaces,
2. **Comparable** with other systems given a way to compare their interfaces and states

Definition

A **systems theory** over the process theory $\mathbb{P}$ is a doubly indexed category with attitude:

$$\mathbf{Sys} : \mathbb{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}^*$$

*More precisely, this is a right module of $\mathbb{P}$, considered as a (pseudo)monad in $\mathbb{S}\mathbf{pan}(\mathbf{Cat})$, since we can't really exchange loose and tight arrows (thanks to DJM for clarifying this).

Systems theories

Crucially, **systems** are

1. **Extendable** by processes going out of their interfaces,
2. **Comparable** with other systems given a way to compare their interfaces and states

Definition

A **systems theory** over the process theory $\mathbb{P}$ is a doubly indexed category with attitude:

$$\mathbf{Sys} : \mathbb{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}^*$$

Again, one can consider more structure, e.g. **concurrent systems theories**, which are systems theories over concurrent process theories (monoidal dbl cats) given by lax monoidal indexings.

*More precisely, this is a right module of $\mathbb{P}$, considered as a (pseudo)monad in $\mathbb{S}\mathbf{pan}(\mathbf{Cat})$, since we can't really exchange loose and tight arrows (thanks to DJM for clarifying this).

Systems theories

The attitude for a systems theory is as follows:

Systems theories

The attitude for a systems theory is as follows:

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

Systems theories

The attitude for a systems theory is as follows:

1. the categories $\mathbf{Sys}(A)$ for a given boundary $A : \mathbb{P}$ are categories of **systems with boundary** A and **simulations** thereof,
2. processes $p : A \mathrel{-\!\!\bullet\!\!\to} B$ act functorially by **extension**:

$$\mathbf{Sys}(p) : \mathbf{Sys}(A) \longrightarrow \mathbf{Sys}(B)$$

Systems theories

The attitude for a systems theory is as follows:

1. the categories $\mathbf{Sys}(A)$ for a given boundary $A : \mathbb{P}$ are categories of **systems with boundary** A and **simulations** thereof,
2. processes $p : A \mathrel{-\!\!\bullet\!\!\to} B$ act functorially by **extension**:

$$\mathbf{Sys}(p) : \mathbf{Sys}(A) \longrightarrow \mathbf{Sys}(B)$$

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

$$\mathbf{Sys}(h) : \mathbf{Sys}(A) \mathrel{\longrightarrow\!\!\!\!\!\!\!+\;\;} \mathbf{Sys}(C)$$

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

Systems theories

4. maps of processes

$$\begin{array}{ccc} A & \xrightarrow{h} & C \\ {\scriptstyle p}\big\downarrow\bullet & \overset{\sigma}{\Longrightarrow} & \bullet\big\downarrow{\scriptstyle q} \\ B & \xrightarrow[k]{} & D \end{array}$$

yield squares

$$\begin{array}{ccc} \mathbf{Sys}(A) & \xrightarrow{\mathbf{Sys}(p)} & \mathbf{Sys}(B) \\ {\scriptstyle \mathbf{Sys}(h)}\big\downarrow & \overset{\mathbf{Sys}(\sigma)}{\Longrightarrow} & \big\downarrow{\scriptstyle \mathbf{Sys}(k)} \\ \mathbf{Sys}(C) & \xrightarrow[\mathbf{Sys}(q)]{} & \mathbf{Sys}(D) \end{array}$$

which **extend** a given simulation of systems along the given map of systems σ.

Example: observable dynamical systems

Traditionally, a **dynamical system** (DS) (Giunti and Mazzola 2012) is an endomorphism

$$\mathrm{update} : S \to S$$

in some category of spaces $\mathbf{S}$ (e.g. $\mathbf{Smooth}$, $\mathbf{Meas}$, ...)

Example: observable dynamical systems

Traditionally, a **dynamical system** (DS) (Giunti and Mazzola 2012) is an endomorphism

$$\mathrm{update} : S \to S$$

in some category of spaces $\mathbf{S}$ (e.g. $\mathbf{Smooth}$, $\mathbf{Meas}$, ...)

An **observable dynamical system** (ODS) (Rosen 1978) is a DS equipped with an observation:

$$\mathsf{S} = (S : \mathbf{S}, \mathrm{update} : S \to S, \mathrm{observe} : S \to O)$$

Example: observable dynamical systems

Traditionally, a **dynamical system** (DS) (Giunti and Mazzola 2012) is an endomorphism

$$\mathrm{update} : S \to S$$

in some category of spaces $\mathbf{S}$ (e.g. $\mathbf{Smooth}$, $\mathbf{Meas}$, ...)
An **observable dynamical system** (ODS) (Rosen 1978) is a DS equipped with an observation:

$$\mathsf{S} = (S : \mathbf{S}, \mathrm{update} : S \to S, \mathrm{observe} : S \to O)$$

We can arrange ODS in a system theory, over a slightly trivial process theory, $\mathbf{S}^{\to}$:

$$\begin{array}{ccc} O & \xrightarrow{h} & O' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle q} \\ Q & \xrightarrow[k]{} & Q' \end{array}$$

Depending on the context, one might make vertical and horizontal maps different, e.g. vertical maps might be effectful while horizontal ones aren't.

Example: observable dynamical systems

Then

$$\mathbf{DynSys} : \mathbf{S}^{\rightarrow} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

$$\mathbf{DynSys}(O) = \left\{ \begin{array}{ccc} S & \xrightarrow{g} & R \\ {\scriptstyle \mathrm{update}_{\mathsf{S}}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ S & \xrightarrow{g} & R \\ {\scriptstyle \mathrm{observe}_{\mathsf{S}}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ O & = & O \end{array} \right\}$$

Example: observable dynamical systems

Then

$$\mathbf{DynSys} : \mathbf{S}^{\rightarrow} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

$$\mathbf{DynSys}(O) = \left\{ \begin{array}{ccc} S & \xrightarrow{g} & R \\ {\scriptstyle \mathrm{update}_{\mathsf{S}}} \downarrow & & \downarrow {\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ S & \xrightarrow{g} & R \\ {\scriptstyle \mathrm{observes}_{\mathsf{S}}} \downarrow & & \downarrow {\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ O & = & O \end{array} \right\}$$

Given a process $p : O \to Q$, we define

$$\mathbf{DynSys}(p) : \mathbf{DynSys}(O) \longrightarrow \mathbf{DynSys}(Q)$$

$$(S, \mathrm{update}_{\mathsf{S}}, \mathrm{observes}_{\mathsf{S}}) \longmapsto (S, \mathrm{update}_{\mathsf{S}}, \mathrm{observes}_{\mathsf{S}} \fatsemi p)$$

Example: observable dynamical systems

Given a map of boundaries $h : O \to O'$, we define a profunctor

$$\mathbf{DynSys}(h) : \mathbf{DynSys}(O)^{\mathsf{op}} \times \mathbf{DynSys}(O') \longrightarrow \mathbf{Set}$$

$$(\mathsf{S}, \mathsf{R}) \longmapsto \left\{ g : S \to R \;\middle|\; \begin{array}{ccc} S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{update}_{\mathsf{S}}}\big\downarrow & & \big\downarrow{\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{observe}_{\mathsf{S}}}\big\downarrow & & \big\downarrow{\scriptstyle \mathrm{update}_{\mathsf{R}}} \\ O & \underset{h}{\longrightarrow} & O' \end{array} \right\}$$

Example: observable dynamical systems

Finally, given a square $\begin{array}{ccc} O & \xrightarrow{h} & O' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle q} \\ Q & \xrightarrow[k]{} & Q' \end{array}$ we define a square

$$\begin{array}{ccc} \mathbf{DynSys}(O) & \xrightarrow{\mathbf{DynSys}(p)} & \mathbf{DynSys}(Q) \\ {\scriptstyle \mathbf{DynSys}(h)}\big\downarrow\!\!\!+ & \xRightarrow{\mathbf{DynSys}(\square)} & +\!\!\!\big\downarrow{\scriptstyle \mathbf{DynSys}(k)} \\ \mathbf{DynSys}(O') & \xrightarrow[\mathbf{DynSys}(q)]{} & \mathbf{DynSys}(Q') \end{array}$$

'sending $g : S \to R$ to itself', with the proof of commutativity got by stacking squares vertically:

$$\begin{array}{ccc} S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{update}_\mathsf{S}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_\mathsf{R}} \\ S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{observe}_\mathsf{S}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_\mathsf{R}} \\ O & \xrightarrow[h]{} & O' \end{array} \quad\longmapsto\quad \begin{array}{ccc} S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{update}_\mathsf{S}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_\mathsf{R}} \\ S & \overset{g}{\dashrightarrow} & R \\ {\scriptstyle \mathrm{observe}_\mathsf{S}}\downarrow & & \downarrow{\scriptstyle \mathrm{update}_\mathsf{R}} \\ O & \xrightarrow[h]{} & O' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle q} \\ Q & \xrightarrow[k]{} & Q' \end{array}$$

Example: Moore machines

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

$$\mathbf{Moore} : \mathbb{L}\mathbf{ens}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

A Moore machine $\mathsf{S} : \mathbf{Moore}\binom{I}{O}$ is an **open dynamical systems**:

$$\mathrm{update}_{\mathsf{S}} : S \times I \longrightarrow S, \mathrm{observe}_{\mathsf{S}} : S \longrightarrow O$$

It's not just open **towards** the outside (*observable*), it's also open **from** the outside.

Example: Moore machines

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

$$\mathbf{Moore} : \mathbb{L}\mathbf{ens}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

A Moore machine $\mathsf{S} : \mathbf{Moore}\binom{I}{O}$ is an **open dynamical systems**:

$$\text{update}_{\mathsf{S}} : S \times I \longrightarrow S, \text{observe}_{\mathsf{S}} : S \longrightarrow O$$

It's not just open **towards** the outside (*observable*), it's also open **from** the outside.

Morphisms are maps that commute with the dynamics

$$\mathbf{Moore}\binom{I}{O} = \left\{ \begin{array}{ccc} \binom{S}{S} & \overset{\pi_2 ⨾ g}{\underset{g}{\rightrightarrows}} & \binom{R}{R} \\ \text{update}_{\mathsf{S}}\uparrow\downarrow\text{observe}_{\mathsf{S}} & & \text{update}_{\mathsf{R}}\uparrow\downarrow\text{observe}_{\mathsf{R}} \\ \binom{I}{O} & = & \binom{I}{O} \end{array} \right\}$$

Example: Moore machines

Given a lens $\begin{pmatrix} p^\sharp \\ p \end{pmatrix} : \begin{pmatrix} I \\ O \end{pmatrix} \rightrightarrows \begin{pmatrix} I' \\ O' \end{pmatrix}$ we get a functor:

$$\mathbf{Sys}\begin{pmatrix} p^\sharp \\ p \end{pmatrix} : \mathbf{Sys}\begin{pmatrix} I \\ O \end{pmatrix} \longrightarrow \mathbf{Sys}\begin{pmatrix} I' \\ O' \end{pmatrix}$$

$$\begin{array}{ccc}
\begin{pmatrix} S \\ S \end{pmatrix} & & \begin{pmatrix} S \\ S \end{pmatrix} \\
 & & \text{update}_S \uparrow\downarrow \text{observe}_S \\
\text{update}_S \uparrow\downarrow \text{observe}_S & \longmapsto & \begin{pmatrix} I \\ O \end{pmatrix} \\
 & & p \downarrow\uparrow p^\sharp \\
\begin{pmatrix} I \\ O \end{pmatrix} & & \begin{pmatrix} I' \\ O' \end{pmatrix}
\end{array}$$

Example: Moore machines

Given a chart $\begin{pmatrix} h^\flat \\ h \end{pmatrix} : \begin{pmatrix} I \\ O \end{pmatrix} \rightrightarrows \begin{pmatrix} J \\ Q \end{pmatrix}$ we get a profunctor:

$$\mathbf{Moore}\begin{pmatrix} I \\ O \end{pmatrix}^{\mathsf{op}} \times \mathbf{Moore}\begin{pmatrix} J \\ Q \end{pmatrix} \xrightarrow{\mathbf{Moore}\begin{pmatrix} h^\flat \\ h \end{pmatrix}} \mathbf{Set}$$

$$\left(\begin{array}{c} \begin{pmatrix} S \\ S \end{pmatrix} \\ \mathrm{update}_{\mathsf{S}} \upharpoonleft\downharpoonright \mathrm{observe}_{\mathsf{S}} \\ \begin{pmatrix} I \\ O \end{pmatrix} \end{array} , \begin{array}{c} \begin{pmatrix} R \\ R \end{pmatrix} \\ \mathrm{update}_{\mathsf{R}} \upharpoonleft\downharpoonright \mathrm{observe}_{\mathsf{R}} \\ \begin{pmatrix} J \\ Q \end{pmatrix} \end{array} \right) \longmapsto \left\{ \begin{array}{ccc} \begin{pmatrix} S \\ S \end{pmatrix} & \overset{\pi_2 ⨾ g}{\underset{g}{\dashrightarrow}} & \begin{pmatrix} R \\ R \end{pmatrix} \\ \mathrm{update}_{\mathsf{S}} \upharpoonleft\downharpoonright \mathrm{observe}_{\mathsf{S}} & & \mathrm{update}_{\mathsf{R}} \upharpoonleft\downharpoonright \mathrm{observe}_{\mathsf{R}} \\ \begin{pmatrix} I \\ O \end{pmatrix} & \overset{h^\flat}{\underset{h}{\rightrightarrows}} & \begin{pmatrix} J \\ Q \end{pmatrix} \end{array} \right\}$$

Example: Moore machines

Finally, given a square

$$
\begin{array}{ccc}
\begin{pmatrix} I \\ O \end{pmatrix} & \overset{h^\flat}{\underset{h}{\rightrightarrows}} & \begin{pmatrix} J \\ Q \end{pmatrix} \\
{\scriptstyle p^\sharp}\uparrow\downarrow{\scriptstyle p} & & {\scriptstyle q^\sharp}\uparrow\downarrow{\scriptstyle q} \\
\begin{pmatrix} I' \\ O' \end{pmatrix} & \overset{k^\flat}{\underset{k}{\rightrightarrows}} & \begin{pmatrix} J' \\ Q' \end{pmatrix}
\end{array}
$$

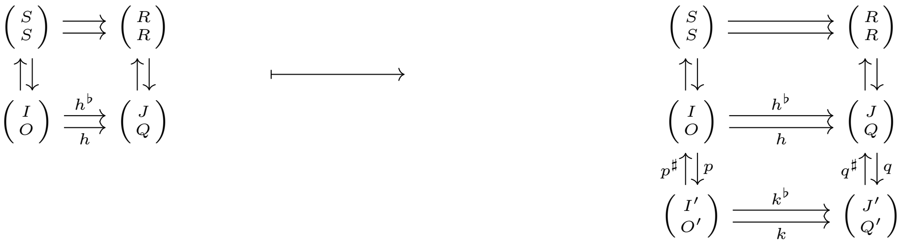
we get a square in $\mathbb{C}\mathbf{at}$...

$$
\begin{array}{ccc}
\mathbf{Moore}\begin{pmatrix} I \\ O \end{pmatrix} & \xrightarrow{\mathbf{Moore}\begin{pmatrix} p^\sharp \\ p \end{pmatrix}} & \mathbf{Moore}\begin{pmatrix} I' \\ O' \end{pmatrix} \\
{\scriptstyle \mathbf{Moore}\begin{pmatrix} h^\flat \\ h \end{pmatrix}}\big\downarrow & \overset{\mathbf{Moore}(\square)}{\Longrightarrow} & \big\downarrow{\scriptstyle \mathbf{Moore}\begin{pmatrix} k^\flat \\ k \end{pmatrix}} \\
\mathbf{Moore}\begin{pmatrix} J \\ Q \end{pmatrix} & \xrightarrow[\mathbf{Moore}\begin{pmatrix} q^\sharp \\ q \end{pmatrix}]{} & \mathbf{Moore}\begin{pmatrix} J' \\ Q' \end{pmatrix}
\end{array}
$$

Example: Moore machines

…given by stacking squares:

$$\mathbf{Moore}\begin{pmatrix} h^\flat \\ h \end{pmatrix}(\mathsf{S}, \mathsf{R}) \xrightarrow{\mathbf{Moore}(\square)_{\mathsf{S},\mathsf{R}}} \mathbf{Moore}\begin{pmatrix} k^\flat \\ k \end{pmatrix}\left(\mathbf{Moore}\begin{pmatrix} p^\sharp \\ p \end{pmatrix}(\mathsf{S}),\ \mathbf{Moore}\begin{pmatrix} q^\sharp \\ q \end{pmatrix}(\mathsf{R})\right)$$



Example: observational theory

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

$$\mathbf{Obs}_{\mathbf{C}} : \mathbb{S}\mathbf{pan}(\mathbf{C})^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

$$\mathbf{Obs}_{\mathbf{C}}(I) = \left\{ \begin{array}{ccc} S & \xrightarrow{h} & R \\ {\scriptstyle \mathrm{observe}_{\mathsf{S}}}\downarrow & & \downarrow{\scriptstyle \mathrm{observe}_{\mathsf{R}}} \\ I & = & I \end{array} \right\}$$

In this theory, a system S over the interface $I : \mathbf{C}$ is simply a state space $S : \mathbf{C}$ together with an observation $\mathrm{observe}_{\mathsf{S}} : S \to I$.

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

Example: observational theory

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

$$\mathbf{Obs}(I \xleftarrow{p_\ell} X \xrightarrow{p_r} I') : \mathbf{Obs}(I) \longrightarrow \mathbf{Obs}(I')$$

$$\begin{array}{ccc} S & & p_\ell^* S \\ & & \downarrow{\scriptstyle p_\ell^* \mathrm{observes}_S} \\ {\scriptstyle \mathrm{observes}_S}\downarrow & \longmapsto & X \\ & & \downarrow{\scriptstyle p_r} \\ I & & I' \end{array}$$

Example: observational theory

Given a map $I \xrightarrow{h} J$, we define a profunctor:

$$\mathbf{Obs}_{\mathbf{C}}(I)^{\mathsf{op}} \times \mathbf{Obs}_{\mathbf{C}}(J) \xrightarrow{\mathbf{Obs}_{\mathbf{C}}(h)} \mathbf{Set}$$

$$\begin{array}{c} S \\ \text{observes}_{\mathsf{S}} \downarrow \\ I \end{array} \qquad \begin{array}{c} R \\ \text{observer}_{\mathsf{R}} \downarrow \\ J \end{array} \longmapsto \left\{ \begin{array}{ccc} S & \longrightarrow & R \\ \text{observes}_{\mathsf{S}} \downarrow & & \downarrow \text{observer}_{\mathsf{R}} \\ I & \xrightarrow{h} & J \end{array} \right\}$$

Example: observational theory

Finally, we have a map on squares that sends

$$
\begin{array}{ccc}
I & \xrightarrow{h} & J \\
{\scriptstyle p_\ell}\uparrow & & \uparrow{\scriptstyle q_\ell} \\
X & \overset{\sigma}{\longrightarrow} & Y \\
{\scriptstyle p_r}\downarrow & & \downarrow{\scriptstyle q_r} \\
I' & \xrightarrow[k]{} & J'
\end{array}
\quad \longmapsto \quad
\begin{array}{ccc}
\mathbf{Obs}_{\mathbf{C}}(I) & \xrightarrow{\mathbf{Obs}_{\mathbf{C}}(p)} & \mathbf{Obs}_{\mathbf{C}}(J) \\
{\scriptstyle \mathbf{Obs}_{\mathbf{C}}(h)}\downarrow & \overset{\mathbf{Obs}_{\mathbf{C}}(\sigma)}{\Longrightarrow} & \downarrow{\scriptstyle \mathbf{Obs}_{\mathbf{C}}(k)} \\
\mathbf{Obs}_{\mathbf{C}}(I') & \xrightarrow[\mathbf{Obs}_{\mathbf{C}}(q)]{} & \mathbf{Obs}_{\mathbf{C}}(J')
\end{array}
$$

again by stacking:

$$
\mathbf{Obs}_{\mathbf{C}}(h)(\mathsf{S},\mathsf{R}) \xrightarrow{\mathbf{Obs}_{\mathbf{C}}(\sigma)_{\mathsf{S},\mathsf{R}}} \mathbf{Obs}_{\mathbf{C}}(k)(\mathbf{Obs}_{\mathbf{C}}(f)(\mathsf{S}),\mathbf{Obs}_{\mathbf{C}}(g)(\mathsf{R}))
$$

$$
\begin{array}{ccc}
S & \longrightarrow & R \\
{\scriptstyle \mathrm{observe}_{\mathsf{S}}}\downarrow & {\scriptstyle \mathrm{observe}_{\mathsf{R}}} & \downarrow \\
I & \xrightarrow[h]{} & J
\end{array}
\quad \longmapsto \quad
\begin{array}{ccc}
S & \longrightarrow & R \\
\downarrow & & \downarrow \\
I & \xrightarrow[h]{} & J \\
\uparrow & & \uparrow \\
X & \overset{\sigma}{\longrightarrow} & Y \\
\downarrow & & \downarrow \\
I' & \xrightarrow[k]{} & J'
\end{array}
$$

Example: observational theory

Thus we have a systems theory:

$$\mathbf{Obs}_{\mathbf{C}} : \mathbb{S}\mathbf{pan}(\mathbf{C})^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

In particular, we have

$$\mathbf{Obs}_{\mathbf{Set}} : \mathbb{S}\mathbf{et}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

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

When we write $\mathbf{Obs}$, this is what we mean.

Morphisms of system theories

In category theory, we are interested in the way things map into each other...

Definition

A **morphism of system theories** $\Phi : \mathbf{Sys} \to \mathbf{Sys}'$ is a vertical lax natural transformation:

$$\begin{array}{ccc} \mathbb{P}^{\top} & \xrightarrow{\Phi^{\top}} & \mathbb{P}'^{\top} \\ {\scriptstyle \mathbf{Sys}}\downarrow & \overset{\Phi^{\flat}}{\Longrightarrow} & \downarrow{\scriptstyle \mathbf{Sys}'} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

Morphisms of system theories

In category theory, we are interested in the way things map into each other...

Definition

A **morphism of system theories** $\Phi : \mathbf{Sys} \to \mathbf{Sys}'$ is a vertical lax natural transformation:

$$\begin{array}{ccc} \mathbb{P}^\top & \xrightarrow{\Phi^\top} & \mathbb{P}'^\top \\ {\scriptstyle \mathbf{Sys}}\big\downarrow & \overset{\Phi^\flat}{\Longrightarrow} & \big\downarrow{\scriptstyle \mathbf{Sys}'} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

Categorical system theory studies system theories

$$\mathbf{Sys} : \mathbb{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

through **behaviours**:

$$B : \mathbf{Sys} \longrightarrow \mathbf{Obs}$$

Corepresentable behaviours

Construction

Given $\mathbf{Sys} : \mathbb{P}^\top \to \mathbb{C}\mathbf{at}$, and a system $\mathsf{S} : \mathbf{Sys}(I)$, there is a morphism:

$$\begin{array}{ccc} \mathbb{P}^\top & \xrightarrow{\mathbb{P}_h(I,-)^\top} & \mathbb{S}\mathbf{et}^\top \\ {\scriptstyle \mathbf{Sys}}\downarrow & \overset{\mathbf{Sys}(\mathsf{S},-)}{\Longrightarrow} & \downarrow{\scriptstyle \mathbf{Obs}} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

Corepresentable behaviours

Construction

Given $\mathbf{Sys} : \mathbb{P}^\top \to \mathbb{C}\mathbf{at}$, and a system $\mathsf{S} : \mathbf{Sys}(I)$, there is a morphism:

$$\begin{array}{ccc} \mathbb{P}^\top & \xrightarrow{\mathbb{P}_h(I,-)^\top} & \mathbb{S}\mathbf{et}^\top \\ {\scriptstyle \mathbf{Sys}}\downarrow & \overset{\mathbf{Sys}(\mathsf{S},-)}{\Longrightarrow} & \downarrow{\scriptstyle \mathbf{Obs}} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

where, for a given $J : \mathbb{P}$,

$$\mathbf{Sys}(\mathsf{S}, -)_J : \mathbf{Sys}(J) \longrightarrow \mathbf{Set}^{\mathbb{P}_h(I,J)}$$

is a functor equivalently given as

$$\begin{array}{ccc} \mathbf{Sys}(J) \times \mathbb{P}_h(I,J) & \longrightarrow & \mathbf{Set} \\ (\mathsf{T}, \quad I \xrightarrow{h} J) & \mapsto & \mathbf{Sys}(h)(\mathsf{S},\mathsf{T}) \end{array}$$

This picks all simulations of S in T mediated by the boundary map h.

Example: fixpoints behaviours

Let fix : $\binom{1}{1} \Longrightarrow \binom{1}{1}$ be the trivial Moore machine (only one states, no input output).

Then $\mathbf{Moore}(\text{fix}, -)$ is the **behaviour of fixpoints**: given another Moore machine T : $\binom{I}{O}$, and a chart $\binom{i}{o} : \binom{1}{1} \rightrightarrows \binom{I}{O}$, the set

$$\mathbf{Moore}\binom{i}{o}(\text{fix}, \mathsf{T}) = \left\{ t \in T \;\middle|\; \begin{array}{ccc} \binom{1}{1} & \overset{\pi_2 \fatsemi s}{\underset{s}{\Longrightarrow}} & \binom{T}{T} \\ \Vert & & {\scriptstyle p^\sharp}\uparrow\downarrow{\scriptstyle p} \\ \binom{1}{1} & \overset{i}{\underset{o}{\Longrightarrow}} & \binom{I}{O} \end{array} \right\} = \{t \in T \mid p(s) = o,\ p^\sharp(t, i) = t\}$$

is the set of fixpoints for input i (and giving output o).

Example: fixpoint behaviours

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

$$\mathbf{Moore}(\mathsf{fix}, -) : \mathbf{Moore} \longrightarrow \mathbf{Obs}$$

In particular, we can automatically mint lots of compositional structure. For instance, we know for any lens $\begin{pmatrix} f^\sharp \\ f \end{pmatrix} : \begin{pmatrix} I \\ O \end{pmatrix} \leftrightarrows \begin{pmatrix} J \\ Q \end{pmatrix}$ there is a natural map

$$\begin{array}{ccc}
\mathbf{Moore}\begin{pmatrix} I \\ O \end{pmatrix} & \xrightarrow{\mathbf{Moore}(\mathsf{fix},-)_I} & \mathbf{Set}^{O\times I} \\
{\scriptstyle \mathbf{Moore}\begin{pmatrix} f^\sharp \\ f \end{pmatrix}} \downarrow & \swArrow {\scriptstyle \mathbf{Moore}\left(\mathsf{fix}, \begin{pmatrix} f^\sharp \\ f \end{pmatrix}\right)} & \downarrow {\scriptstyle f_*} \\
\mathbf{Moore}\begin{pmatrix} J \\ Q \end{pmatrix} & \xrightarrow[\mathbf{Moore}(\mathsf{fix},-)_J]{} & \mathbf{Set}^{Q\times J}
\end{array}$$

sending fixpoints of $\mathsf{S} : \mathbf{Moore}\begin{pmatrix} I \\ O \end{pmatrix}$ to fixpoints of its extension $\mathbf{Moore}\begin{pmatrix} f^\sharp \\ f \end{pmatrix}(\mathsf{S})$.

And one can prove in this case the map is iso! This recovers (D. I. Spivak 2015), and generalizes more (see (Myers 2021, Theorem 5.3.3.1)).

Recap: categorical systems theory

1. One starts by defining a **process theory**, i.e. a double category with attitude:

$$\mathbb{P} := \left\{ \begin{array}{ccc} \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \\ {\scriptstyle\text{process}}\big\downarrow & \overset{\text{map of processes}}{\Longrightarrow} & \big\downarrow{\scriptstyle\text{process}} \\ \cdot & \xrightarrow{\text{map of interfaces}} & \cdot \end{array} \right\}$$

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

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$$

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

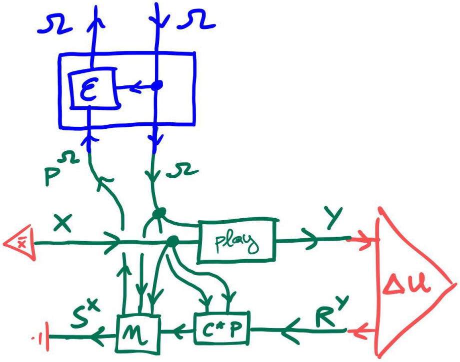
$$\begin{array}{ccc} \mathbb{P}^{\top} & \xrightarrow{B^{\top}} & \mathbb{S}\mathbf{et}^{\top} \\ {\scriptstyle\mathbf{Sys}}\big\downarrow & \overset{B^{\flat}}{\Longrightarrow} & \big\downarrow{\scriptstyle\mathbf{Obs}} \\ \mathbb{C}\mathbf{at} & = & \mathbb{C}\mathbf{at} \end{array}$$

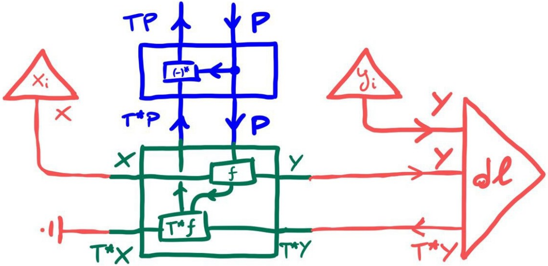
We get many of these just a corepresentable ones, i.e. maps of type $\mathbf{Sys}(\mathsf{S}, -)$.

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 Capucci, Gavranović, Hedges, and Rischel 2021 and in Smithe 2021 rests on two main pillars:

Categorical cybernetics

The conceptual foundation of ‘categorical cybernetics’, as advocated in Capucci, Gavranović, Hedges, and Rischel 2021 and in Smithe 2021 rests on two main pillars:

1. The fact cybernetic systems are *mereologically* peculiar in having a distinctive boundary between ‘**controller**’ and ‘**controlled**’ subsystems;

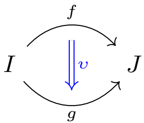
Categorical cybernetics

The conceptual foundation of 'categorical cybernetics', as advocated in Capucci, Gavranović, Hedges, and Rischel 2021 and in Smithe 2021 rests on two main pillars:

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

The compositional structure of cybernetic processes

The basic component of a bicategory of cybernetic processes is a globular 2-cell:



The ‘cybernetic’ interpretation of such a globe is that I and J denote an interface, f and g are controllable processes between the given interfaces, and v ‘reduces’ the controllability of g to that of f, being itself a dynamical process.

The compositional structure of cybernetic processes

Their composition algebra also checks out: **vertical composition of globular cells models sequential composition of controls**, whereas horizontal composition models the 'parallel' composition of controls arising from the sequential composition of the processes they control.

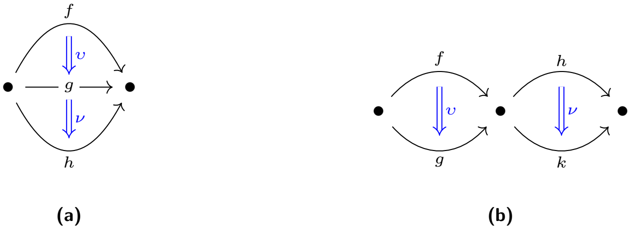
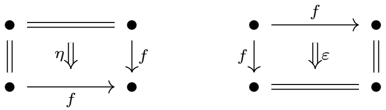


Figure: *(a) sequential composition of controls, (b) parallel composition of controls.*

The compositional structure of cybernetic processes

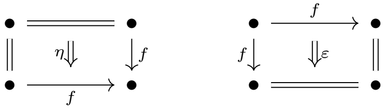
With little effort, we can concert controllable and simple processes in the same structure. We claim the structure of controllable processes is that a of a **company** a double category where every tight arrow has a companion.



Idea: simple processes are trivially controllable processes, so 'appear both tightly and loosely'.

The compositional structure of cybernetic processes

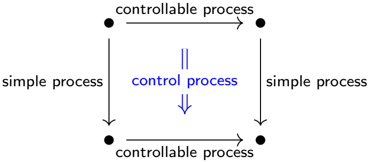
With little effort, we can concert controllable and simple processes in the same structure. We claim the structure of controllable processes is that a of a **company** a double category where every tight arrow has a companion.



Idea: simple processes are trivially controllable processes, so 'appear both tightly and loosely'.

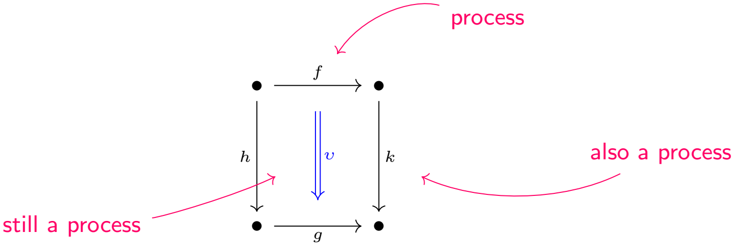
Definition

A **cybernetic company** is a a company with attitude whose objects are **interfaces**, loose arrows are **controllable processes**, tight arrows are **simple processes**, and squares are **control processes**:



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.



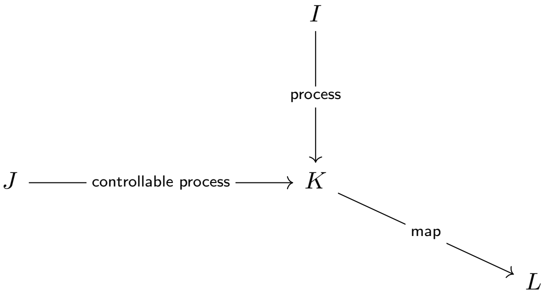
Thus we are forced to go 3D!

Cybernetic process theories

Definition

A **cybernetic process theory** is a triple category with attitude $\mathfrak{P}$ 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**,

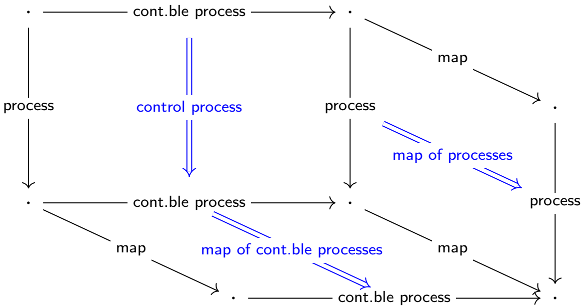


Cybernetic process theories

Definition

A **cybernetic process theory** is a triple category with attitude $\mathfrak{P}$ where

1. frontal 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**

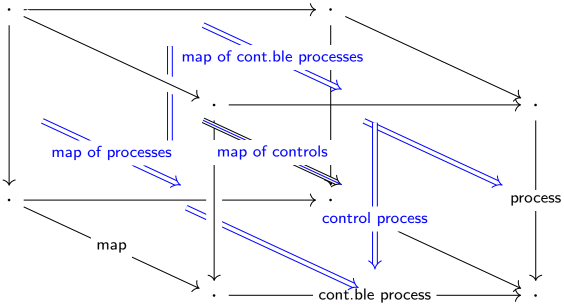


Cybernetic process theories

Definition

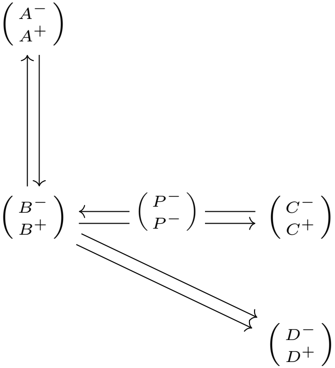
A **cybernetic process theory** is a triple category with attitude $\mathfrak{P}$ where

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



Example: parametric lenses

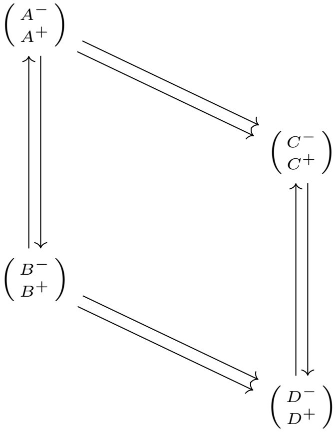
One can promote the process theory of lenses to a cybernetic theory $\mathbb{P}\mathbf{ara}(\mathbb{L}\mathbf{ens})$[1], with parametric lenses in the role of controllable processes:



[1] This notation is actually sound: there's a Para construction that yields this triple category

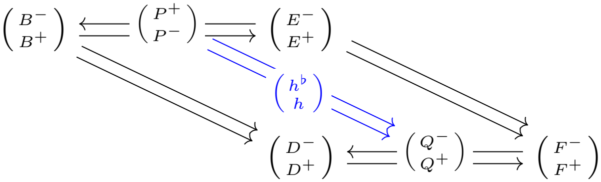
Example: parametric lenses

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



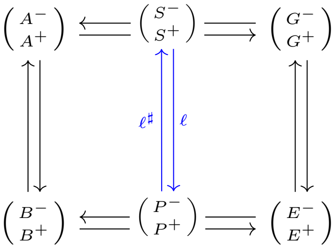
Example: parametric lenses

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



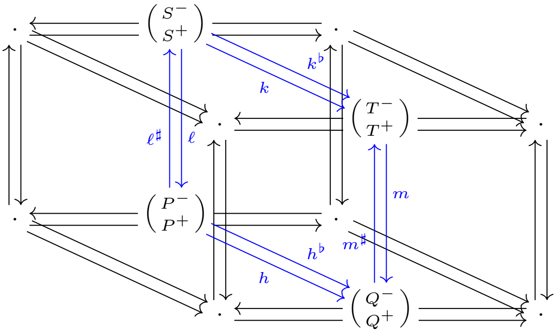
Example: parametric lenses

Basic 2-cells are commutative squares of parametric lenses:



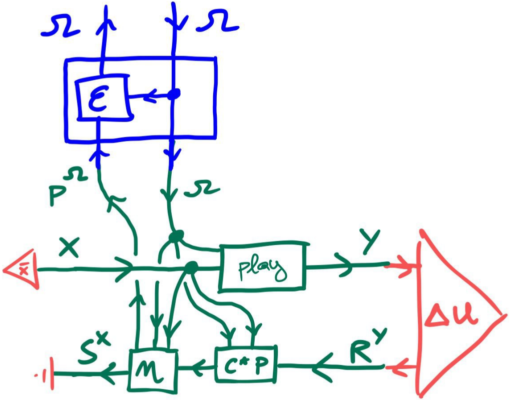
Example: parametric lenses

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



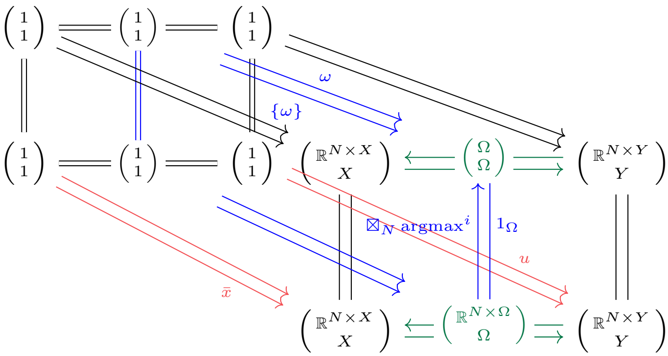
Example: Nash equilibria

Categorical game theory (Capucci 2022; Capucci, Gavranović, Hedges, and Rischel 2021; Ghani, Hedges, Winschel, and Zahn 2018) is nowadays based on the idea games are parametric lenses, controlled by players.



Example: Nash equilibria

Categorical game theory is nowadays based on the idea games are parametric lenses, controlled by players. In the same way fixpoints of Moore machines are maps (squares) from fix into them, one can show **Nash equilibria** of games are maps (cubes) from fix into them:



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

Towards behavioural cybernetics

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

Towards behavioural cybernetics

We would like to recover the compositional formulae for Nash equilibria as functoriality of some corepresentable behaviour “$\mathbf{Plrs}(\text{fix}, -)$”, 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. An **observational cybernetic system theory**
3. A notion of **behaviour functor** between the two

Towards behavioural cybernetics

We would like to recover the compositional formulae for Nash equilibria as functoriality of some corepresentable behaviour "$\mathbf{Plrs}(\mathrm{fix}, -)$", 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. An **observational cybernetic system theory**
3. A notion of **behaviour functor** between the two

It takes time to unpack all of these, so unfortunately I'm only going to be able to tell you about (1)

Cybernetic system theories

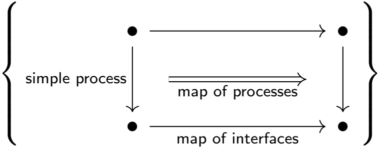
The crucial idea in the definition of cybernetic system theory is that a cybernetic process theory $\mathfrak{P}$ can be decomposed into two process theories:

Cybernetic system theories

The crucial idea in the definition of cybernetic system theory is that a cybernetic process theory $\mathfrak{P}$ can be decomposed into two process theories:

1. A **simple part**, given by the double category of interfaces, maps theoreof, simple processes and maps thereof.

$\mathfrak{P}_t :=$

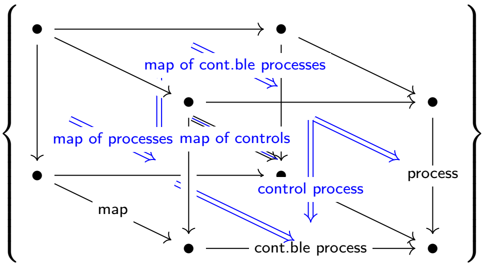


Cybernetic system theories

The crucial idea in the definition of cybernetic system theory is that a cybernetic process theory $\mathfrak{P}$ can be decomposed into two process theories:

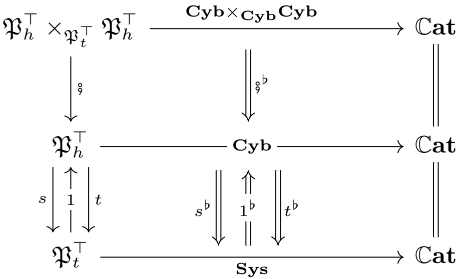
1. A **simple part**, given by the double category of interfaces, maps theoreof, simple processes and maps thereof.
2. A **cybernetic part**, whose objects are cybernetic processes, vertical 1-cells are control processes, horizontal 1-cells are maps of controllable processes, and squares are maps of control processes.

$\mathfrak{P}_h :=$



Cybernetic system theory

This suggests one can define a cybernetic system theory by putting together a system theory $\mathbf{Sys}$ on $\mathfrak{P}_t$ and a system theory $\mathbf{Cyb}$ on $\mathfrak{P}_h$:



Cybernetic system theory

One can show that $\mathbf{Cyb}$ and $\mathbf{Sys}$ arrange to yield a triple functor

$$\mathbf{CybSys} : \mathfrak{P}^\top \xrightarrow{\text{unitary lax-lax}} \mathbb{S}\mathbf{pan}(\mathbb{C}\mathbf{at})$$

where $(-)^\top$ exchanges vertical and transversal 1-cells.

Thus, together with $s^\flat$ and $t^\flat$, $\mathbf{Cyb}$ assigns a **span of functors** to each controllable process:

Idea: the category $\mathbf{Cyb}$ picks out, for a given controllable process $I \xrightarrow{p} J$, is a category control system U together with simple systems S 'closing off' p. This makes sense: **a cybernetic system is a controllable system coupled to a control system**.

$$I \xrightarrow{p} J \quad \longmapsto \quad \mathbf{Sys}(I) \xleftarrow{s^\flat_p} \mathbf{Cyb}(I \xrightarrow{p} J) \xrightarrow{t^\flat_p} \mathbf{Sys}(J)$$

Cybernetic system theory

The triple functor $\mathfrak{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{S}\mathbf{pan}(\mathbb{C}\mathbf{at})$...

1. ...**associates a category to every controllable process**, which are **reindexed functorially by control processes** and **profunctorially by maps of controllable processes**,

Cybernetic system theory

The triple functor $\mathfrak{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{S}\mathbf{pan}(\mathbb{C}\mathbf{at})$...

1. ...**associates a category to every controllable process**, which are **reindexed functorially by control processes** and **profunctorially by maps of controllable processes**,
2. ...**gives each cybernetic system a source and a target simple system**:

$$s^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(I),$$
$$t^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(J)$$

Cybernetic system theory

The triple functor $\mathfrak{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{S}\mathbf{pan}(\mathbb{C}\mathbf{at})$...

1. ...**associates a category to every controllable process**, which are **reindexed functorially by control processes** and **profunctorially by maps of controllable processes**,
2. ...**gives each cybernetic system a source and a target simple system**:

$$s^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(I),$$
$$t^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(J)$$

3. ...gives a mapping between simple systems with interface I and cybernetic systems controlling the trivial process on I:

$$1^\flat_I : \mathbf{Sys}(I) \longrightarrow \mathbf{Cyb}(I = I)$$

Cybernetic system theory

The triple functor $\mathfrak{P}^\top \xrightarrow{\text{unitary lax}} \mathbb{S}\mathbf{pan}(\mathbb{C}\mathbf{at})$...

1. ...**associates a category to every controllable process**, which are **reindexed functorially by control processes** and **profunctorially by maps of controllable processes**,
2. ...**gives each cybernetic system a source and a target simple system**:

$$s^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(I),$$
$$t^\flat_{I \xrightarrow{p} J} : \mathbf{Cyb}(I \xrightarrow{p} J) \to \mathbf{Sys}(J)$$

3. ...gives a mapping between simple systems with interface I and cybernetic systems controlling the trivial process on I:

$$1^\flat_I : \mathbf{Sys}(I) \longrightarrow \mathbf{Cyb}(I = I)$$

4. ...features a **compatibility structure with sequential composition of cybernetic processes**, given by

$$\fatsemi^\flat_{\substack{I \xrightarrow{p} J \\ J \xrightarrow{q} K}} : \mathbf{Cyb}(I \xrightarrow{p} J) \times_{\mathbf{Sys}(J)} \mathbf{Cyb}(J \xrightarrow{q} K) \longrightarrow \mathbf{Cyb}(I \xrightarrow{p} J \xrightarrow{q} K)$$

Conclusions

Conclusions

There's lots of work to do!

Conclusions

There's lots of work to do!

In categorical system theory, many examples deserve to be explored more. The behavioural study of machines is an big subject, will categorical abstractions prove themselves useful?

Conclusions

There's lots of work to do!

In categorical system theory, many examples deserve to be explored more. The behavioural study of machines is an big subject, will categorical abstractions prove themselves useful?

In categorical cybernetic systems theory, it remains to develop and explore theories of cybernetic systems both in general and particular.

Conclusions

There's lots of work to do!

In categorical system theory, many examples deserve to be explored more. The behavioural study of machines is an big subject, will categorical abstractions prove themselves useful?

In categorical cybernetic systems theory, it remains to develop and explore theories of cybernetic systems both in general and particular.

First in line are applications to game theory, where it can be used to mint compositional structure for equilibrium concepts.

Conclusions

There's lots of work to do!

In categorical system theory, many examples deserve to be explored more. The behavioural study of machines is an big subject, will categorical abstractions prove themselves useful?

In categorical cybernetic systems theory, it remains to develop and explore theories of cybernetic systems both in general and particular.

First in line are applications to game theory, where it can be used to mint compositional structure for equilibrium concepts.

Secondly, how can this framework be useful for other cybernetic systems theories? Can it be used to frutifully map between learning & games? Can we use it to prove theorems across different kinds of systems? (e.g. good regulator theorems/internal model principles/FEP).

Thanks for your attention!

Questions?

References I

- S. Abramsky and B. Coecke, "A categorical semantics of quantum protocols", in *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, IEEE, 2004, pp. 415–425.
- J. C. Baez and K. Courser, "Structured cospans", *Theory and Applications of Categories*, vol. 35, no. 48, pp. 1771–1822, 2020.
- J. C. Baez and J. Master, "Open petri nets", *Mathematical Structures in Computer Science*, vol. 30, no. 3, pp. 314–341, 2020.
- J. C. Baez, K. Courser, and C. Vasilakopoulou, "Structured versus Decorated Cospans", *Compositionality*, vol. 4, 3 Sep. 2022, ISSN: 2631-4444. DOI: 10.32408/compositionality-4-3. [Online]. Available: https://doi.org/10.32408/compositionality-4-3.
- M. Capucci, "Diegetic representation of feedback in open games",, 2022, Available at https://arxiv.org/abs/2206.12338.

References II

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.

B. Coecke and A. Kissinger, "Picturing quantum processes", in *International Conference on Theory and Application of Diagrams*, Springer, 2018, pp. 28–31.

B. Coecke and E. O. Paquette, "Categories for the practising physicist", in *New structures for physics*, Springer, 2010, pp. 173–286.

G. S. H. Cruttwell, B. Gavranović, 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.

E. Di Lavore, A. Gianola, M. Román, N. Sabadini, and P. Sobociński, "A canonical algebra of open transition systems", in *International Conference on Formal Aspects of Component Software*, Springer, 2021, pp. 63–81.

References III

- J. L. Fiadeiro and V. Schmitt, “Structured co-spans: An algebra of interaction protocols”, in *International Conference on Algebra and Coalgebra in Computer Science*, Springer, 2007, pp. 194–208.
- B. Fong, “Decorated cospans”, *Theory & Applications of Categories*, vol. 30, 2015.
- B. Fong, D. Spivak, and R. Tuyéras, “Backprop as functor: A compositional perspective on supervised learning”, in *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, IEEE, 2019, pp. 1–13.
- N. Ghani, J. Hedges, V. Winschel, and P. Zahn, “Compositional game theory”, in *Proceedings of the 33rd annual ACM/IEEE symposium on logic in computer science*, 2018, pp. 472–481.
- A. Gianola, S. Kasangian, and N. Sabadini, “Cospan/span(graph): An algebra for open, reconfigurable automata networks”, in *7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

References IV

- M. Giunti and C. Mazzola, "Dynamical systems on monoids: Toward a general theory of deterministic systems and motion", in *Methods, models, simulations and approaches towards a general theory of change*, World Scientific, 2012, pp. 173–185.
- M. Grandis, *Higher Dimensional Categories: From Double to Multiple Categories*. World Scientific, 2019.
- P. Katis, N. Sabadini, and R. F. Walters, "Bicategories of processes", *Journal of Pure and Applied Algebra*, vol. 115, no. 2, pp. 141–178, 1997.
- ——, "Span(graph): A categorical algebra of transition systems", in *International Conference on Algebraic Methodology and Software Technology*, Springer, 1997, pp. 307–321.
- ——, "Feedback, trace and fixed-point semantics", *RAIRO-Theoretical Informatics and Applications*, vol. 36, no. 2, pp. 181–194, 2002.
- S. Libkind, A. Baas, E. Patterson, and J. Fairbanks, "Operadic modeling of dynamical systems: Mathematics and computation",, 2021, Available at https://arxiv.org/abs/2105.12282.

References V

- D. J. Myers, *Categorical Systems Theory*. 2021, (Work in progress). [Online]. Available: http://davidjaz.com/Papers/DynamicalBook.pdf (visited on 05/14/2022).
- R. Rosen, *Fundamentals of measurement and representation of natural systems*. Elsevier Science Limited, 1978, vol. 1.
- B. Shapiro and D. I. Spivak, "Dynamic categories, dynamic operads: From deep learning to prediction markets",, 2022, Available at https://arxiv.org/abs/2205.03906.
- T. S. C. Smithe, "Compositional active inference i: Bayesian lenses. statistical games", 2021, [Online]. Available: https://arxiv.org/abs/2109.04461.
- D. I. Spivak, "The operad of wiring diagrams: Formalizing a graphical language for databases, recursion, and plug-and-play circuits",, 2013, Available at https://arxiv.org/abs/1305.0297.
- ——, "The steady states of coupled dynamical systems compose according to matrix arithmetic",, 2015, Available at https://arxiv.org/abs/1512.00802.

References VI

D. Vagner, D. I. Spivak, and E. Lerman, "Algebras of open dynamical systems on the operad of wiring diagrams",, 2014, Available at https://arxiv.org/abs/1408.1598.