NOTES ON CATEGORICAL SYSTEMS THEORY

MATTEO CAPUCCI
UNIVERSITY OF STRATHCLYDE

1. Introduction

Systems are ubiquitous, in science as in life. As we look into a thing, we soon realize it is a system comprised of smaller interacting parts,[1]. As we walk back, we realize it is itself part of an even more complex system.

The word 'system' roughly means 'composite thing in Greek. It seems therefore natural that category theory has something to say about them.

Categorical System Theory (CST) is a categorical framework for the abstract study of systems irrespective of their contingent aspects. Instead of espousing a specific paradigm on the mathematical specification of systems, CST predicates upon the general features such paradigms should have. In doing so, it captures the essence of the notion of 'system'.

The object of study of CST is thus a *doctrine of systems*, which is defined in [Mye22] as way to answer the following questions:

- What does it mean to be a system? Does it have a notion of states, or of behaviours? Or is it a diagram describing the way some primitive parts are organized?
- What should the interface of a system be?
- How can interfaces be connected in composition patterns?
- How are systems composed through composition patterns between their interfaces.
- What is a map between systems, and how does it affect their interfaces?
- When can maps between systems be composed along the same composition patterns as the systems.

A doctrine of systems specializes in many different *theories of systems*. For instance there is a doctrine of open dynamical systems encompassing the theory of deterministic dynamical systems, the theory of stochastic dynamical systems, the theory of differential dynamical systems, and many more. Hence it's usually easier to start describing what a theory of systems is and then to say what does it mean for a doctrine to gather many of them in a single object.

1.1. **References.** Categorical approaches to general system theory have been around for a long time, see for example [Ros78], but the current abstract and general form is due to David Jaz Myers, who distilled lots of recent work in an elegant doubly-categorical toolset.

The subject is still in its infancy. At the moment, most of CST lives in Myers' own book [Mye22], itself a longer version of the shorter preprint [Mye20c] (where the notion of *doctrine* wasn't yet developed). Moreover, Myers has given a few talks about the topic in the past years:

[1]Heck, even *atoms*, indivisible par excellence, arise from many layers of interaction among their parts!

(1) D. J. Myers. *A general definition of open dynamical system.* 2020. URL: https://www.youtube.com/watch?v=8T-Km3taNko
(2) D. J. Myers. *Open dynamical systems, trajectories and hierarchical planning.* 2020. URL: https://www.youtube.com/watch?v=3FxeY5DbPn0
(3) D. J. Myers. *Double Categories of Open Dynamical Systems.* 2020. URL: https://www.youtube.com/watch?v=f9fjf9lo2_M
(4) D. J. Myers. *Paradigms of composition.* 2020. URL: https://www.youtube.com/watch?v=50s62D5Ah-M

I also gave a talk about CST and its extension to cybernetic systems:

(5) M. Capucci. *From categorical systems theory to categorical cybernetics.* 2022. URL: https://www.youtube.com/watch?v=wtgfyjFIHBQ

1.2. **A quick tour of CST.** Categorical systems theory is a conceptually simple, if mathematically sophisticated, framework. In a nutshell, it studies processes connecting systems, and the ways these behave. Processes are organized in (monoidal double) categories, which themselves index categories of systems, whose maps in (the observational theory of) sets are behaviours.

If one is not at ease with double categories, at a first approximation one can drop the horizontal direction and think of these as monoidal categories of processes. They index sets of systems which can be reindexed by processes. One can study behaviour by specifying the set of ways interfaces can be observed, the relations processes induce between observations on their interfaces, and the states systems can be in and the observables these expose.

However, none of the two dimensions in CST is ancillary to the other. The horizontal direction is often overlooked in pre-CST work, but it's extremely natural to consider: from a categorical standpoint, we study things (here, systems and processes) by looking at the way they map into each other.

Thus the first step in CST is to understand **processes organize in monoidal double categories**:

$$\mathbb{P} := \left\{ \begin{tikzcd} \cdot \arrow[rr, "\text{map of interfaces}"] \arrow[d, "\text{process}"'] & & \cdot \arrow[d, "\text{process}"] \\ {} \arrow[rr, phantom, "\xRightarrow{\text{map of processes}}"] & & {} \\ \cdot \arrow[rr, "\text{map of interfaces}"'] & & \cdot \end{tikzcd} \right\} \tag{1.1}$$

Such processes are actually 'composition patterns' that can be used to weave systems together, i.e. the ways parts can come together to form wholes. These can be wiring diagrams, or bubble diagrams, or circuit diagrams, etc. Both ways of thinking about them can be useful.

Mathematically speaking, **processes index systems**, giving rise to doubly indexed categories called **systems theories**:

$$\mathbf{Sys} : \mathbb{P}^{\top} \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at} \tag{1.2}$$

Thus, and this is a fundamental idea in CST, systems and processes are formally distinguished, even though they might end up being quite similar. In fact, in many instances, systems are special instances of processes which are considered stateful. The categories of systems over a given interface are categories of structure-preserving morphisms of systems, which we call simulations here. These can be more or less rigid depending on the user's taste.

Finally, systems are as interesting as the things they do. The observations we can make of a system are its behaviour. Ways to observe systems in a given theory are **theories of behaviour**, which are maps into the 'observational theory':

$$\begin{tikzcd} \mathbb{P}^\top \arrow[dr, bend left, "\mathbf{Sys}"] \arrow[d, "B^\top"'] \arrow[dr, phantom, "B^\flat \Downarrow"] & \\ \mathbb{S}\mathbf{et}^\top \arrow[r, bend right, "\mathbf{Set}/-"'] & \mathbb{C}\mathbf{at} \end{tikzcd} \tag{1.3}$$

Here $\mathbb{S}\mathbf{et}$ is the double category of spans in **Set** and $\mathbb{C}\mathbf{at}$ is the double category of functors and profunctors.

1.3. **Prerequisites.** CST is deeply rooted in double category theory. This is dictated by the structure of processes: they compose like morphisms but are also the subject of morphisms.

For many notions, we will reference [Gra19]. For a slow paced, well-motivated introduction of the minimum double category theory used in CST, we invited the reader to read along the main reference [Mye22].

2. Processes

The starting point for defining a theory of systems is defining a double category of composition patterns such systems use to interact with one another. In practice, defining the systems and defining the composition patterns are activities that influence one another. Systems are made out of composition patterns themselves, and it's usually them we have in mind when we approach a formalization problem. So one often starts by asking how the systems at hand could possibly be composed together.

Composition patterns usually form an operad.[2] Operads are indeed ways to specify how 'small things fit into larger things', i.e. they are *theories of composition*. Most importantly, they give meaning to various kinds of wiring diagrams [Spi13; VSL14; LBPF21].

One can also see composition patterns as *processes* that extend a given system with further dynamics, possibly gathering many systems in one. This point of view can be more natural from a 'European' point of view, more acquainted with string diagrams rather than wiring diagrams. In fact one can see double categories of composition patterns as higher-dimensional extensions of the process theories of Abramsky, Coecke, Gogioso [AC04; CK18].

Definition 2.1. A **process theory** is a symmetric monoidal double category with attitude.

Hence any monoidal double category can be a process theory if we have a convincing interpreation of it as such. In other words, at this level of generality there's no justification for limiting this definition further. Of course any specific doctrine of systems can make more opinionated choices on which monoidal double categories to admit, and we will see plenty of examples of this.

[2] By 'operad' we mean 'coloured operad' which means 'multicategory'.

Remark 2.1.1. Rather than defining double categories of processes to be monoidal, one should arguably define them to be multicategories. Thus a nicer, if more exotic, definition of process theory would be that of a 'multicategory internal to categories'.This is the reason here, following the custom recently adopted by Myers, we often denote processes as having many inputs. With our definitions, one might interpret the notation $I_1, \ldots, I_n$ as denoting a monoidal product, i.e. we denote with ',' the monoidal product on processes.

A process theory $\mathbb{P}$ thus unpacks as follows:

(1) There are a number of objects $I, J, K, \ldots$ which are **interfaces** of the processes.
(2) There are horizontal maps $h : I \to J$ which are **maps of interfaces**, without dynamic content, they simply compare different interfaces with each other.[3]

$$I \xrightarrow{h} J \tag{2.1}$$

(3) There are vertical maps $p : I_1, \ldots, I_n \twoheadrightarrow K$ which are the **processes**, connecting interfaces in some way.[4]

$$\begin{tikzcd} I_1, \ldots, I_n \arrow[d, "p"'] \\ K \end{tikzcd} \tag{2.2}$$

(4) There are squares $\alpha : p \Rightarrow q$ which are **maps of processes** supported by given maps of interfaces:

$$\begin{tikzcd} I_1, \ldots, I_n \arrow[r, "h_1, \ldots, h_n"] \arrow[d, "p"'] \arrow[dr, phantom, "\Rightarrow \alpha"] & J_1, \ldots, J_n \arrow[d, "q"] \\ K \arrow[r, "k"'] & L \end{tikzcd} \tag{2.3}$$

A very important takeway behind this definition is that it distinguishes morphisms *as* processes from morphisms *of* processes. In fact, the usual yoga of process theories (which amount to symmetric monoidal 'single' categories) only focuses on the compositional properties of processes (they compose like morphisms). But in category theory we know that if we want to study something, we need to study morphisms between them!

Example 2.1 (Alphabets)**.** Let $\mathbb{A}\mathbf{lph} = \mathbf{FinSet}^{\uparrow}$ be the double category of alphabets and alphabet reductions, and maps thereof. Its objects are finite sets of symbols we call 'alphabets'. Its horizontal maps are maps of finite sets. Its vertical maps are *alphabet reductions*, which are map of finite sets in the opposite direction:

$$p : \Sigma \twoheadrightarrow \Sigma' \in \mathbf{FinSet}(\Sigma', \Sigma) \tag{2.4}$$

[3]One could call these 'algebraic maps of interfaces', as their role is to provide morphisms that *preserve* the interface structure, in order to compare them.

[4]Again, one might call these 'geometric maps of interfaces', as they *reflect* structure in many example we care about

Squares are commutative squares:

$$
\begin{tikzcd}
\Sigma \arrow[r, "h"] & \Xi \\
\Sigma' \arrow[u, "p"] \arrow[r, "k"'] & \Xi' \arrow[u, "q"']
\end{tikzcd}
\tag{2.5}
$$

Example 2.2 (Bidirectional processes)**.** Consider the following double category of *deterministic bidirectional processes*:

(1) interfaces are given by pairs or sets $\binom{A^-}{A^+}$, whose monoidal product is given by component-wise product,
(2) processes $\binom{A^-}{A^+} \leftrightarrows \binom{C^-}{C^+}$ are given by **lenses** $\binom{f^\sharp}{f}$, comprised of a *get part* $f : A^+ \to C^+$ and a *put part* $f^\sharp : A^+ \times C^- \to A^-$,
(3) maps of interfaces $\binom{A^-}{A^+} \rightrightarrows \binom{B^-}{B^+}$ are given by **charts** $\binom{g^\flat}{g}$, comprised of a *states part* $g : A^+ \to B^+$ and a *directions part* $g^\flat : A^+ \times A^- \to B^-$,
(4) behaviours are given by arrangements

$$
\begin{tikzcd}
\binom{A^-}{A^+} \arrow[r, shift left, "g^\flat"] \arrow[r, shift right, "g"'] \arrow[d, shift right, "f"'] & \binom{B^-}{B^+} \arrow[d, shift right, "k"'] \\
\binom{C^-}{C^+} \arrow[r, shift left, "h^\flat"] \arrow[r, shift right, "h"'] \arrow[u, shift right, "f^\sharp"'] & \binom{D^-}{D^+} \arrow[u, shift right, "k^\sharp"']
\end{tikzcd}
\tag{2.6}
$$

such that for every $a^+ \in A^+$ and $c^- \in C^-$:

$$
\begin{aligned}
k(g(a^+)) &= h(f(a^+)), \\
g^\flat(a^+, f^\sharp(a^+, c^-)) &= k^\sharp(g(a^+), h^\flat(f(a^+), c^-)).
\end{aligned}
\tag{2.7}
$$

The last conditions are hard to parse formally, but basically they say that both squares one can spot in (2.6) commute. Concretely, we have two bidirectional processes $\binom{f^\sharp}{f}$ and $\binom{k^\sharp}{k}$ and a way to map between their interfaces. We are then asking that their dynamics commute with such maps.

Remark 2.1.2. Viewed as composition patterns, lenses are algebras of the operad of wiring diagrams [Spi13]. Thus they represent ways to wire a number of boxes into a larger box:

Matteo: wiring diagram

Example 2.3. The previous example can be generalized greatly by employing F-lenses [Spi19]. These are lenses in which the backward part is dependent on the forward part in a way specified by an indexed category F. Intuitively, this correspond to a wiring pattern which can change dependening on the what flows in the wires (see [**poly**]).

The definitions of F-lenses and F-charts are substantially identical to the ones above, as well as that for the squares (except the 'commutativity condition' is now harder to eyeball). We gather some examples here:

	category	F
deterministic	$\mathcal{C}$ cartesian monoidal	$\mathbf{Set}/_{\mathsf{proj}}-$ (or $\mathbf{coKl}(-\times =)$)
possibilistic	$\mathcal{E}$ topos	$\mathbf{biKl}(-\times =, \mathcal{P})$ ($\mathcal{P}$ powerset monad)
probabilistic	**Msbl**	$\mathbf{biKl}(-\times =, \Delta)$ (Δ probability monad)
effectful	$\mathcal{C}$ cartesian monoidal	$\mathbf{biKl}(-\times =, M)$ (M commutative monad)
differential (Euclidean)	**Euc**	$\mathbf{Euc}/_{\mathsf{subm}}-$
differential (general)	**Smooth**	$\mathbf{Smooth}/_{\mathsf{subm}}-$

Example 2.4. Consider the following double category of *variable-sharing processes*:

(1) interfaces are variables, embodied by a set A of their values,
(2) processes are functional maps of variables, embodied by functions $f : A \to C$ between their values,
(3) maps of interfaces are variable sharing patterns, embodied by spans $g : A \leftarrow R \to B$ between their values,
(4) behaviours are squares in $\mathbf{Span}(\mathbf{Set})$, hence commutative diagrams such that:

$$\begin{tikzcd} A \arrow[d, "f"'] & R \arrow[l, "g_\ell"'] \arrow[r, "g_r"] \arrow[d, "\sigma"] & B \arrow[d, "k"] \\ C & S \arrow[l, "h_\ell"] \arrow[r, "h_r"'] & D \end{tikzcd} \tag{2.8}$$

hence such that for each $r \in R$,

$$\begin{aligned} f(g_\ell(r)) &= h_\ell(\sigma(r)), \\ h_r(\sigma(r)) &= k(g_r(r)). \end{aligned} \tag{2.9}$$

Example 2.5. Given any Cartesian category[5] spans & maps

3. Systems

Systems are the things processes link, or the things composition patterns compose.

Definition 3.1. A **doubly indexed category**, or **action of a double category**, is given, informally, by a unitary lax double functor $\mathbf{Sys} : \mathbb{P}^\top \to \mathbb{C}\mathbf{at}$. If $\mathbb{P}$ is monoidal, then we also ask $\mathbf{Sys}$ to be lax monoidal.

We recall $\mathbb{C}\mathbf{at}$ is the cartesian monoidal double category of categories, functors, profunctors and natural transformations. A unitary lax functor is a double functor preserving vertical identities strictly but not vertical composition. Hence given two maps of interfaces $h : I \to J$, $k : J \to K$, we have a natural transformation $\ell_{h,k} : \mathbf{Sys}(h) \otimes \mathbf{Sys}(k) \to \mathbf{Sys}(h \mathbin{⨟} k)$ between profunctors.

Definition 3.2. A **theory of systems over the process theory** $\mathbb{P}$ is a monoidal doubly indexed category $\mathbf{Sys} : \mathbb{P}^\top \to \mathbb{C}\mathbf{at}$ with attitude.

[5]By which we mean a category with a terminal object and all pullbacks.

Concretely, **Sys** maps interfaces to **categories of systems**, processes to **extension functors**, maps of interfaces to **mapping profunctors** and maps of processes to **extension transformations**.

Hence given an interface $I : \mathbb{P}$, we think of the objects of $\mathbf{Sys}(I)$ as systems of a certain kind while the maps are **simulations** between them, i.e. some notion of structure-preserving map between them.

$$\mathbf{Sys}(I) = \left\{ \ \mathsf{S} \xrightarrow{\varphi} \mathsf{T} \ \right\} \tag{3.1}$$

The functors induced by a process act by extending a system with that process. If we think of the process as a composition pattern instead, the functor assembles in a composite system:

$$\mathbf{Sys}(I \overset{p}{\bullet\!\!\to} K) : \mathbf{Sys}(I) \longrightarrow \mathbf{Sys}(K) \tag{3.2}$$

The profunctors induced by a map of interfaces give notions of simulations between systems on different interfaces:

$$\mathbf{Sys}(I \xrightarrow{h} J) : \mathbf{Sys}(I) \nrightarrow \mathbf{Sys}(J) \tag{3.3}$$

Hence an element $\ell \in \mathbf{Sys}(I \xrightarrow{h} J)(\mathsf{S}, \mathsf{T})$ is a *simulation of* S *in* T *mediated by the maps of interfaces* h.

Finally, squares in $\mathbb{P}$ induce squares witnessing the extension of a simulation of systems along a map of processes:

$$\mathbf{Sys}(p \overset{\alpha}{\Rightarrow} q) : \mathbf{Sys}(h) \Rightarrow (\mathbf{Sys}(p), \mathbf{Sys}(q))^* \mathbf{Sys}(k) \tag{3.4}$$

Example 3.1 (Closed dynamical systems)**.** The most basic model of dynamical systems in mathematics is simply endomorphisms $S : X \to X$ on some space X in a category of 'spaces' $\mathcal{S}$. These systems are closed: they expose nothing of their state, and their dynamics can't be influenced by external input: their process theory is trivial! Consequently, the systems theory of closed dynamical systems is given by a single category $\mathbf{DynSys}_{\mathcal{S}}$ picked out by a functor $1 \to \mathbb{C}\mathbf{at}$. In this category, the objects are endomorphisms and the maps are commuting squares of the form:

$$\begin{tikzcd} X \arrow[r, "\varphi"] \arrow[d, "S"'] & Y \arrow[d, "R"] \\ X \arrow[r, "\varphi"'] & Y \end{tikzcd} \tag{3.5}$$

Clearly $\mathbf{DynSys}_{\mathcal{S}}$ is monoidal if $\mathcal{S}$ is. Thus given a category $\mathcal{S}$ of spaces, one gets a systems theory of **closed dynamical systems** in $\mathcal{S}$.

Example 3.2. There's many possible variations on the definition of $\mathbf{DynSys}_{\mathcal{S}}$. Two that encompass many interesting examples are as follows.

(1) One can choose an endofunctor $F : \mathcal{S} \to \mathcal{S}$ and consider F-coalgebras instead of mere endomorphisms as the dynamical systems. In this way one can get, e.g. non-deterministic closed systems. We denote this category $\mathbf{Coalg}(F)$. The basic case is recovered for the choice $F = 1_{\mathcal{S}}$.

(2) One can choose a monoid of 'time' T and consider T-actions instead of mere endomorphisms. Notice one can pick $T : \mathbf{Mon}(\mathcal{S})$ but also $T : \mathbf{Mon}(\mathbf{Set})$, and then consider T-actions to be functors $BT \to \mathcal{S}$. In this way one can get, e.g. continuous time dynamical systems by choosing $T = \mathbb{R}$. We denote this category $\mathbf{TimeSys}(T)$. The basic case is recovered for the choice $T = \mathbb{N}$.

These two examples also admit a more interesting theory of processes, as we are going to see shortly.

Example 3.3. We can think of a coalgebra $A \to FA$ as system with states A and interface F. Now, natural transformations $\alpha : F \Rightarrow F'$ are 'lenses' and one gets an indexed category

$$\mathbf{Coalg} : \mathrm{End}(\mathcal{C}) \to \mathbf{Cat} \tag{3.6}$$

Now, if $\mathcal{C}$ is additionally finitely complete, we can go further and add another dimension. In fact, in this case, $\mathrm{End}(\mathcal{C})$ is fibred over $\mathcal{C}$ by evaluation at the terminal object:

$$-(1) : \mathrm{End}(\mathcal{C}) \to C \tag{3.7}$$

The cartesian lift of a given arrow $f : A \to G(1)$ is given by a natural transformation $f_G : f^*G \Rightarrow G$ obtained from the pullback square

$$\begin{tikzcd} f^*GX \arrow[r, "f_{G,X}"] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & GX \arrow[d, "G!"] \\ A \arrow[r, "f"'] & G1 \end{tikzcd} \tag{3.8}$$

that simultaneously defines f^*G (on morphisms is defined by pullback again) and f_G.

The fibred subcategory of polynomial functors is what gives $\mathcal{C}/-$-lenses, whose opposite is the codomain fibration, i.e. $\mathcal{C}/-$-charts. This suggests that taking the opposite fibration of $-(1)$ gives us a fibration of 'generalized charts'.

We can explicitly construct these things if we work out the cartesian factorization system induced by $-(1)$ on $\mathrm{End}(\mathcal{C})$. This is given by

(1) Cartesian maps are given by natural transformations whose naturality is witness by pullback squares, as suggested by the definition of f^*G above: which we make explicit here:

$$\begin{tikzcd} FX \arrow[r, "\alpha_X"] \arrow[d, "Ff"'] \arrow[dr, phantom, "\lrcorner", very near start] & GX \arrow[d, "Gf"] \\ FY \arrow[r, "\alpha_Y"'] & GY \end{tikzcd} \tag{3.9}$$

(2) Vertical maps are given by natural transformations whose component at 1 is an isomorphism (think: the identity)

We define a *generalized chart* $(k, k^\flat) : F \rightrightarrows G$ to be a span in $\mathrm{End}(\mathcal{C})$ whose left leg is vertical and whose right leg is cartesian. Hence they look like this:

$$\begin{tikzcd} \mathrm{End}(\mathcal{C}) \arrow[d, "-(1)"'] & F & f^*G \arrow[l, "k^\flat"'] \arrow[r, "k_G"] & G \\ \mathcal{C} & F1 \arrow[r, equal] & F1 \arrow[r, "k"'] & G1 \end{tikzcd} \tag{3.10}$$

These might look like lenses (because lenses are obtained by opping a fibration) but they are actually charts. We can verify this by looking at the case in which F and G are polynomial, to see if this construction recovers the usual one. We see that $k^\flat$ lives in

$$\mathrm{Nat}\left(\sum_{i \in F1} y^{F_i}, \sum_{j \in F1} y^{G_{k(j)}}\right) \cong \prod_{i \in F1} \sum_{j \in F1} F_i \to G_{k(j)} \tag{3.11}$$

and since we know this is a vertical map, meaning it lives over the identity map of $F1$, we know $i = j$ on the right hand side. Thus we see $(k, k^\flat)$ encodes the data of a chart (notice how F and G swapped places: charts are lenses 'relative to lenses').

This allows us to extend the indexed category **Coalg** defined previously to have a profunctorial action. So a given generalized chart $(k, k^\flat) : F \rightrightarrows G$ is mapped to a profunctor $\mathbf{Coalg}(k, k^\flat) : \mathbf{Coalg}(F) \nrightarrow \mathbf{Coalg}(G)$. This has a rather complex definition: it maps two coalgebras $\gamma : A \to FA$ and $\delta : B \to GB$ to the set of $\phi : A \to B$ that make the following commute:

$$\begin{tikzcd} A \arrow[d, "\gamma"'] \arrow[rr, dashed, "\langle \gamma; F!, \phi \rangle"] \arrow[rrrr, bend left, "\phi"] && k^*B \arrow[rr, "k'"] \arrow[d, "\delta'"] \arrow[drr, phantom, "\lrcorner"] && B \arrow[d, "\delta"] \\ FA \arrow[d, "F(\phi)"'] \arrow[drr, "F!"] && && \\ FB && f^G B \arrow[ll, "k^\flat_B"'] \arrow[rr] \arrow[d, "G!'"'] \arrow[drr, phantom, "\lrcorner"] && GB \arrow[d, "G!"] \\ && F1 \arrow[rr, "k"'] && G1 \end{tikzcd} \tag{3.12}$$

$$k^\flat_B(\delta'(\langle \gamma; F!, \phi \rangle(a))) = F(\phi; k')(\gamma(a))$$

Example 3.4 (Monoid actions)**.** The categories $\mathbf{TimeSys}(T)$ naturally gather in a doubly indexed category where the indexing double category is $\mathbf{Mon}(\mathcal{S})^\uparrow$. This is the double category of monoids in $\mathcal{S}$ and commutative squares thereof, except we take the opposite of the vertical direction. Hence a vertical morphism $p : M \twoheadrightarrow N$ corresponds to a morphism of monoids $N \to M$.

This double category is a process theory for the theory of dynamical systems 'with time' described above. In fact a vertical morphism $p : M \twoheadrightarrow N$ maps to a functor $\mathbf{TimeSys}(M) \to \mathbf{TimeSys}(N)$ given by 'restriction of scalars':

$$M \times X \xrightarrow{S} X \qquad \longmapsto \qquad N \times X \xrightarrow{p \times X} M \times X \xrightarrow{S} X, \tag{3.13}$$

and maps of monoids $h : M \to N$ also map to profunctors $\mathbf{TimeSys}(M) \nrightarrow \mathbf{TimeSys}(N)$ that sends a pair of dynamical systems $S : \mathbf{TimeSys}(M)$, $R : \mathbf{TimeSys}(N)$ to the set of squares

$$\begin{tikzcd} M \times X \arrow[r, dashed, "h\times\varphi"] \arrow[d, "S"'] & N \times Y \arrow[d, "R"] \\ X \arrow[r, dashed, "\varphi"'] & Y \end{tikzcd} \tag{3.14}$$

Example 3.5 (Labelled transition systems)**.** We can use the double category $\mathbb{A}\mathbf{lph}$ of alphabets defined in Example 2.1 to index labelled transition systems. When $T = \Sigma^*$ (the free monoid on a finite set Σ), then $\mathbf{DynSys}_{\mathcal{S}}(\Sigma^*)$ is the category of transition systems labelled in Σ. Hence we define $\mathbf{LabTransSys} : \mathbb{A}\mathbf{lpha}^\top \xrightarrow{\text{unitary lax}} \mathbb{C}\mathbf{at}$ by restricting $\mathbf{DynSys}_{\mathbf{Set}}$ along the double functor $(-)^* : \mathbf{Mon}^\uparrow \to \mathbb{A}\mathbf{lph}$ given by taking free monoids.

Example 3.6. Moore machines

Example 3.7. Mealy machines

Example 3.8. coalgebras

Example 3.9. observational systems

Example 3.10. structured cospans

3.1. **Doctrines.** A doctrine of systems is a uniform way to specify theories of systems given some data. Most of the theories we described above are actually doctrines, since we defined them parametric on some data:

(1) a theory of observational systems is defined for each Cartesian category $\mathcal{C}$,
(2) a theory of Moore machines is defined for every fibration $\pi : \mathcal{E} \to \mathcal{C}$ with a section T,
(3) a theory of coalgebras is defined for every category $\mathcal{C}$,

and so on.

Definition 3.3. A **doctrine of systems** is a 2-functor into $\mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}$.

The 2-category $\mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}$ has system theories as objects and the following as 1-cells:

Definition 3.4. A **map of system theories** is a pair $(F, F^\flat) : \mathbf{Sys}_1 \to \mathbf{Sys}_2$ where F is a lax double functor while $F^\flat$ is a vertical lax-natural transformation.

$$\begin{tikzcd} \mathbb{P}_1^\top \arrow[dr, bend left, "\mathbf{Sys}_1"] \arrow[dd, "F^\top"'] & \\ \arrow[r, phantom, "F^\flat\Downarrow"] & \mathbb{C}\mathbf{at} \\ \mathbb{P}_2^\top \arrow[ur, bend right, "\mathbf{Sys}_2"'] & \end{tikzcd} \tag{3.15}$$

The 2-cells in $\mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}$ are pairs of an horizontal natural transformation and a modification [Gra19].

4. Behaviours

Behaviours in CST are simply maps into an observational theory. By default, we consider behaviours valued in $\mathbf{Obs}(\mathbf{Set})$, but one can consider 'enriched behaviours' into $\mathbf{Obs}(\mathcal{C})$ when one wants to keep track of extra structure on the set of behaviours of a system.

Definition 4.1. A **theory of behaviour** for a system theory **Sys** valued in the Cartesian category $\mathcal{C}$ is a map of system theories $B : \mathbf{Sys} \to \mathbf{Obs}(\mathcal{C})$:

$$\begin{tikzcd} \mathbb{P}^\top \arrow[drr, bend left, "\mathbf{Sys}"] \arrow[d, "B^\top"'] & \arrow[d, phantom, "B^\flat\Downarrow"] & \\ \mathbf{Span}(\mathcal{C})^\top \arrow[rr, bend right, "\mathcal{C}/-"'] & & \mathbb{C}\mathbf{at} \end{tikzcd} \tag{4.1}$$

A **doctrine of behaviour** for the doctrine $\mathbb{D}\mathbf{octrine}$ is map of doctrines of systems into the observational doctrine **Obs**:

$$\begin{tikzcd} \mathbb{D}\mathbf{ata} \arrow[drr, bend left, "\mathbb{D}\mathbf{octrine}"] \arrow[d, "\mathbb{B}^\top"'] & \arrow[d, phantom, "\mathbb{B}^\flat\Downarrow"] & \\ \mathbb{C}\mathbf{art}\mathbb{C}\mathbf{at} \arrow[rr, bend right, "\mathbf{Obs}"'] & & \mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h} \end{tikzcd} \tag{4.2}$$

Remark 4.1.1. The laxity of $B^\flat$ relates the behaviours of the parts of a system to the behaviour of the whole system. The non-invertibility of such a map witnesses emergent behaviours. In [Mye22, Theorem 5.3.3.1], Myers proves that a large class of behaviours for the doctrine of Moore machines does not, in fact, exhibit emergence, by showing such the laxity of $B^\flat$ is invertible.

Example 4.1. A large class of behaviours are corepresentables, i.e. defined by simulations of an archetypal system exhibiting that behaviour. Thus there is a **doctrine of corepresentable behaviour** on the doctrine of pointed theories:

$$\begin{tikzcd} \mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}_* \arrow[drr, bend left, "U"] \arrow[d, "\mathbf{Set}"'] & \arrow[d, phantom, "\mathrm{Hom}^h\Downarrow"] & \\ \mathbb{C}\mathbf{art}\mathbb{C}\mathbf{at} \arrow[rr, bend right, "\mathbf{Obs}"'] & & \mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h} \end{tikzcd} \tag{4.3}$$

An object of $\mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}_*$ is a system theory equipped with a distinguished system (hence a pair of an interface and a system over it) which is used as archetype for a certain kind of behaviour.

The transformation Hom^h at a pointed theory $(\mathbf{Sys} : \mathbb{P}^\top \to \mathbb{C}\mathbf{at}, \mathsf{B} : \mathbf{Sys}(I))$ is the map of system theories $\mathbb{P}^h(\mathsf{B}, -) : \mathbf{Sys} \to \mathbf{Obs}(\mathbf{Set})$ defined as follows. Its two components are the horizontal hom-functor $\mathbb{P}^h(I, -) : \mathbb{P} \to \mathbb{S}\mathbf{et}$ and the similar fiberwise hom-functor

$$\mathbb{P}^h(\mathsf{B}, -)^\flat : \mathbf{Sys} \Rightarrow \mathbf{Set}/\mathbb{P}^h(I, -). \tag{4.4}$$

This latter functor sends a system $\mathsf{S} : \mathbf{Sys}(J)$ to the $\mathbb{P}^h(I, J)$-indexed family of sets sending a map of interfaces $k : I \to J$ to the set $\mathbf{Sys}(k)(\mathsf{B}, \mathsf{S})$ of maps of systems mediated by k.

Example 4.2. Let $\mathbb{M}\mathbf{oore}\mathbf{Data}^!$ be the 2-category of fibrations of categories with a terminal object and a section thereof. These amount to a fibration $p : \mathcal{E} \to \mathcal{C}$ such that $p(1_\mathcal{E}) = 1_\mathcal{C}$, and a section $T : \mathcal{C} \to \mathcal{E}$ such that $T(1_{\mathcal{C}}) = 1_{\mathcal{E}}$. In this situation, one can build the Moore machine $\mathsf{fix} : \binom{T1}{1} \leftrightharpoons \binom{1}{1}$ which 'does nothing'. Hence we get a *doctrine of fixpoints* by using such a machine as the archetype for the behaviour of a still system:

$$\begin{tikzcd}
\mathbb{M}\mathbf{ooreData}^{!} \arrow[d, "{(\mathbf{Moore},\ \mathsf{fix}:\mathbf{Moore}(1))}"'] \arrow[ddr, bend left, "\mathbf{Moore}"] & \\
\mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h}_{*} \arrow[dr, "U"] \arrow[dd, "\mathbf{Set}"'] \arrow[d, phantom, shift left=6ex, "\mathrm{Hom}^h \Downarrow"] & \\
& \mathbb{S}\mathbf{ys}\mathbb{T}\mathbf{h} \\
\mathbb{C}\mathbf{artCat} \arrow[ur, "\mathbf{Obs}"'] &
\end{tikzcd} \tag{4.5}$$

References

[AC04] 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.

[Cap22] M. Capucci. *From categorical systems theory to categorical cybernetics*. 2022. URL: https://www.youtube.com/watch?v=wtgfyjFIHBQ.

[CK18] B. Coecke and A. Kissinger. "Picturing quantum processes". In: *International Conference on Theory and Application of Diagrams*. Springer. 2018, pp. 28–31.

[Gra19] M. Grandis. *Higher Dimensional Categories: From Double to Multiple Categories*. World Scientific, 2019.

[LBPF21] S. Libkind et al. "Operadic modeling of dynamical systems: mathematics and computation". In: (2021). Available at https://arxiv.org/abs/2105.12282.

[Mye20a] D. J. Myers. *A general definition of open dynamical system*. 2020. URL: https://www.youtube.com/watch?v=8T-Km3taNko.

[Mye20b] D. J. Myers. *Double Categories of Open Dynamical Systems*. 2020. URL: https://www.youtube.com/watch?v=f9fjf9lo2_M.

[Mye20c] D. J. Myers. "Double categories of open dynamical systems". In: *arXiv preprint arXiv:2005.05956* (2020).

[Mye20d] D. J. Myers. *Open dynamical systems, trajectories and hierarchical planning*. 2020. URL: https://www.youtube.com/watch?v=3FxeY5DbPn0.

[Mye20e] D. J. Myers. *Paradigms of composition*. 2020. URL: https://www.youtube.com/watch?v=50s62D5Ah-M.

[Mye22] D. J. Myers. *Categorical Systems Theory*. 2022.

[Ros78] R. Rosen. *Fundamentals of measurement and representation of natural systems*. Vol. 1. Elsevier Science Limited, 1978.

[Spi13] D. I. Spivak. "The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits". In: (2013). Available at https://arxiv.org/abs/1305.0297.

[Spi19] D. I. Spivak. "Generalized Lens Categories via functors $F : \mathcal{C}^{\mathrm{op}} \to \mathsf{Cat}$". 2019.

[VSL14] D. Vagner, D. I. Spivak, and E. Lerman. "Algebras of open dynamical systems on the operad of wiring diagrams". In: (2014). Available at https://arxiv.org/abs/1408.1598.