Massachusetts Institute of Technology

Dynamical Systems and Sheaves

DYNAMICAL SYSTEMS AND SHEAVES

PATRICK SCHULTZ, DAVID I. SPIVAK, AND CHRISTINA VASILAKOPOULOU

ABSTRACT. A categorical framework for modeling and analyzing systems in a broad sense is proposed. These systems should be thought of as 'machines' with inputs and outputs, carrying some sort of signal that occurs through some notion of time. Special cases include continuous and discrete dynamical systems (e.g. Moore machines). Additionally, morphisms between the different types of systems allow their translation in a common framework. A central goal is to understand the systems that result from arbitrary interconnection of component subsystems, possibly of different types, as well as establish conditions that ensure totality and determinism compositionally. The fundamental categorical tools used here include lax monoidal functors, which provide a language of compositionality, as well as sheaf theory, which flexibly captures the crucial notion of time.

CONTENTS

1. INTRODUCTION

The broad goal of this paper is to suggest a solid categorical framework for understanding and simulating systems of systems. While the current work is certainly theoretical, our ultimate interest is to make the case that category theory can be useful for modeling real-world systems, certainly not a novel thesis as we discuss in the related work section. We consider broadly defined *open* (dynamical) systems that take in, process, and send out material. Their complexity level varies greatly; for example, they can be used to model anything from an electric circuit, to a chemistry experiment, to a robot. Moreover, they are designed to be interconnected: the material output by one system is sent to, and received by, another. The central idea to which the current work adheres, building on earlier

(PS) BROAD INSTITUTE, 415 MAIN ST, CAMBRIDGE, MA 02142, USA

(DS) MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 77 MASSACHUSETTS AVE. CAMBRIDGE, MA 02139

(CV) UNIVERSITY OF CALIFORNIA, RIVERSIDE, 900 UNIVERSITY AVENUE, 92521, USA

E-mail addresses: pschultz@broadinstitute.org, dspivak@mit.edu, vasilak@ucr.edu.

works like [RS13, Spi13, VSL15], is that a single system may arise by wiring together any number of component open subsystems; see Fig. 1. Analyzing a composite system is often
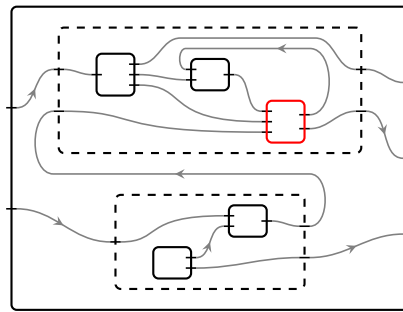

FIGURE 1. Compositional analyses facilitate the rearrangement as well as the replacement of internal components

intractable, because its complexity is generally exponential in the number of subsystems. Hence it is often crucial that the analysis be *compositional* [SY13] because such analyses can be applied to the subsystems independently, and the results can be composed in a specific sense. This also means that the analysis is robust to redesign: improvements can arise from reconfiguring any one of the numerous parts and subparts of large-scale systems (see Fig. 1) at any level of a hierarchy, which can itself be re-structured, and the analyses of unaffected systems remain valid.

The theory of monoidal categories and operads provides an excellent formalism in which to study compositionality. In our approach, an object in the symmetric monoidal category $\mathcal{W}_{\mathcal{C}}$ of $\mathcal{C}$-*labeled boxes* and *wiring diagrams* (for some category $\mathcal{C}$) looks like a box in Fig. 1, thought of as an interface for an open dynamical system with input and output ports through which it can interact with its environment. A morphism in $\mathcal{W}_{\mathcal{C}}$ describes how systems can be interconnected together to form new systems: for example, any of the two dotted composite systems in Fig. 1 are morphisms in the underlying operad, whereas that picture constitutes an operad composition of a 3-ary (top) and a 2-ary (bottom) morphism producing a 5-ary one. This framework is a cornerstone for the present work and will be described in detail; wiring diagrams become the algebraic operations for combining dynamical systems.

The *typing* category $\mathcal{C}$ indicates the sort of information or material that each component in a system may send or receive. For example, if $\mathcal{C} = \mathbf{Set}$, each port is associated to a set of possible inputs/outputs signals. To properly model the flow of such signals, we must decide whether changes are happening continuously, at discrete time steps, or in some combination; that is how the fundamental issue of time enters our formalization. In particular, for a discrete dynamical system the flow occurs on the ticks of a global clock, whereas for continuous dynamical systems it occurs continuously throughout intervals of time. Furthermore, when different types of systems are composed, their notions of time must be faithfully translated into a common one, keeping in mind that there is no 'best' notion of time: the more expressive a language is for describing systems, the more difficult it is to answer questions about them.

The categories, or more precisely toposes, that capture the required variations of time in this paper are sheaves on intervals in $\mathbb{R}$ or in $\mathbb{N}$, structures also studied in [Law86]. Such typing categories, herein denoted $\widetilde{\mathbf{Int}}$ or $\widetilde{\mathbf{Int}_N}$, endow input and output ports with sets of allowed trajectories of information over each (discrete or continuous) interval of time: these trajectories can restrict over smaller intervals of time, and compatible ones can glue to produce a trajectory over the union of intervals. There is also a (slice) topos $\widetilde{\mathbf{Int}}/\mathrm{Sync}$ encompassing 'synchronized continuous time', in which each continuous interval is assigned a phase $\theta \in [0, 1)$. If one wishes to combine continuous-time systems with discrete ones that operate on the same clock, the result is in the topos of *synchronous* sheaves; see Remark 3.3.1.

Having established the interface and the time framework, our world of interacting open systems is expressed as a *wiring diagram algebra*, namely a lax monoidal functor $F \colon \mathcal{W}_{\mathcal{C}} \to \mathbf{Cat}$ or equivalently an operad morphism from the underlying operad. For any such functor and X a $\mathcal{C}$-labeled box, FX is the category of F-systems (for example, Moore machines or continuous dynamical systems) with input and output ports determined by the shape of X. Given F-systems 'inhabiting' the boxes of a picture like Fig. 1, the lax monoidal structure along with functoriality of F coherently produce a composite F-system that inhabits the outer box. Moreover, any functor $H \colon \mathcal{C} \to \mathcal{D}$ induces a strong monoidal functor $\mathcal{W}_H \colon \mathcal{W}_{\mathcal{C}} \to \mathcal{W}_{\mathcal{D}}$ capable of changing the input/output types. A morphism, then, of wiring diagram algebras is a *monoidal opindexed 1-cell* [MV18], i.e. a monoidal natural transformation

$$
\begin{array}{ccc}
\mathcal{W}_{\mathcal{C}} & \xrightarrow{\ F\ } & \\
\downarrow{\scriptstyle \mathcal{W}_H} & \Downarrow \alpha & \mathbf{Cat} \\
\mathcal{W}_{\mathcal{D}} & \xrightarrow[\ G\ ]{} &
\end{array}
$$

with components $\alpha_X \colon FX \to G(\mathcal{W}_H(X))$ that specifically translate F-systems to G-systems.

Our work is based on a class of very general, span-like wiring diagram algebras, which we simply call *machines*. A lax monoidal functor $\mathsf{Spn}_{\mathcal{C}} \colon \mathcal{W}_{\mathcal{C}} \to \mathbf{Cat}$ (Proposition 2.4.1) gives rise to certain open systems called *discrete*, *continuous*, or *synchronous* machines, corresponding to the situation in which the typing category $\mathcal{C}$ is the topos of discrete, continuous, or synchronous interval sheaves respectively. Not only do ordinary discrete and continuous dynamical systems translate into discrete and continuous machines via an embedding of algebras, but also they can ultimately compose to one another since all machines embed into the algebra of synchronous ones, $\mathsf{Mch}_{\mathrm{Sync}} \colon \mathcal{W}_{\widetilde{\mathbf{Int}}/\mathrm{Sync}} \to \mathbf{Cat}$.

A very important feature of our machines is that—in all time versions—there exist subalgebras of *inertial*, *deterministic*, and *total* machines. In more detail, certain lifting conditions examine whether a system can determine a piece of a future output trajectory, whether a state could evolve in more than one way, or whether a state could potentially 'die', when the current state is subject to some input trajectory. The fact that such subcategories form algebras themselves means that the respective definitions are carefully chosen so as to be closed under arbitrary compositions, including feedback. For that, the formalism of $\mathcal{W}_{\mathcal{C}}$-algebras is essential; such a result would not hold if traced monoidal categories were used instead, roughly for the same reason that [ABP99] introduces trace ideals: "there appears to be a tension between having identities and having compact closed structure", or

in our terminology, identity maps are not inertial. Finally, for all above machine variations, there exist *contracted machines* defined as systems of the same type that comply with a given 'contract', expressed as a sub-presheaf. A logic of behavior contracts in a slightly different setting was later given in [SS17].

Related work and future goals. The present work is part of a far more general, ongoing endeavor by many researchers to categorically formalize general processes and exhibit the advantages of such compositional analyses. Our results aim to contribute towards a better understanding of systems, using the theory of monoidal categories to capture nesting and sheaf theory to capture time. In particular, as mentioned earlier, the topos $\widetilde{\mathbf{Int}}$ of sheaves on real intervals was also studied by Lawvere in [Law86] for similar purposes; this becomes evident especially in Appendix A. Notably, all of our models of time assume a common reference time frame for all components in the system: real time progresses for everyone at the same rate. If one wants a more flexible model for non-interacting parallel subsystems (i.e. concurrency), or to allow for relativistic effects, **Int**-sheaves would have to be replaced by a different formalism such as event structures [Win87] or action structures [Mil96].

The parts of this work oriented to discrete dynamical systems connect to the well-established theory of composing finite state automata (Moore machines) or more generally cyber-physical systems, see e.g. [LS17]. In our framework, the basic operations are feedback and parallel placement, expressed as morphisms and monoidal product in the wiring diagram category; for example, serial composition (or CASCADE) is derived from those. An advantage of this framework is that every wiring diagram arrangement of automata, like the one in Fig. 1, can be directly given a precise mathematical formulation as their composite. This constitutes a universal way of composing systems, agnostic to the complications of interconnections; we plan to further pursue this point—as well as compositions of Moore machines with different types of systems—according to current challenges in the respective fields.

Our general system notion is essentially a span inside $\widetilde{\mathbf{Int}}$, which is in fact equivalent to a *discrete Conduché fibration* [Joh99] as explained in Appendix A.3. In [Fio00, BF00], the authors present a very similar strategy of expressing a general process via a discrete Conduché fibration (therein called *unique factorization lifting* functor) over monoids of intervals, also distinguishing between discrete and continuous variations. They also discuss *hybrid* systems which can in fact be modeled as continuous machines, see [SS17]. Thus there exist strong similarities between the core formalisms of the two works which should be further investigated, while also examining certain differences: first, the distinction between input and output for us is essential—we are interested in the conditions under which deterministic and total systems compose—whereas their systems communicate with the outer world via a single *control* space. Moreover, the two works' scopes differ a great deal: we focus on nesting and composing systems of different types, whereas they focus on the designs of languages and their logic.

Such an investigation is also relevant to connections of the current framework to a more behavioral approach to systems, e.g. [WP13], which abandons the input/output distinction: our span formalization of machines is already symmetric in that sense. This is

also related to machines giving an immediate instance of a *hypergraph category*, recently shown to be equivalent to algebras for an operad of cospans [FS18]. On the other hand, total machines in our context seem to be strongly associated to the concept of *open maps* in the theory of bisimulation, see e.g. [JNW96], where again the distinction between input and output is critical.

Our work is also connected to the theory of traced monoidal categories [JSV96] and PROPs in the following way. Recall that a *prop* is a symmetric strict monoidal category whose monoid of objects is $(\mathbb{N}, 0, +)$, the free monoid on one generator. A *wheeled prop* (see e.g. [MMS09]) is a traced, symmetric, strict monoidal category whose underlying (cancellable) monoid of objects is again $(\mathbb{N}, 0, +)$. The diagrams in a wheeled prop look very similar to diagrams in this paper like (11), so it is worth elaborating on their differences. First of all, it was shown in [SSR16] that the category of wheeled props is equivalent to that of algebras on the operad 1-**CatCob** of oriented 1-cobordisms. The operad of wiring diagrams does indeed compare to that of cobordisms, but the two are not equivalent. In more detail, in a wheeled prop there is only one generating object, but we have a generating object for each sheaf; thus one should begin by using a colored wheeled prop. Moreover, wires in our diagrams are allowed to split or terminate; thus one should assume that each color in the prop is equipped with a comonoid structure. Finally, wires in our diagrams are not allowed to feed straight through; this is a sort of *dialectica* condition in the sense of [Pai90], which renders it unreasonable to ask for a map from cobordisms to $\mathcal{W}$ since the usual axiomatization of traced monoidal categories depends on the identity. Thus while it is true that every wheeled colored prop with a comonoid structure on every object induces a wiring diagram algebra in our sense, the converse does not hold.

Our work is also similar in spirit to that of [KSW00] on systems with boundary. This work builds on a notion of a *category with feedback*, closely related to traced monoidal categories but with a built-in notion of delay, used to define an *ifo* (input-feedback-delay) system: examples of those include circuits, algorithms as well as Mealy machines. A significant example is that of spans of graphs, which are shown to model discrete time processes; since graphs are precisely sheaves on discrete intervals $\widetilde{\mathbf{Int}}_N$ as explained by Proposition 3.2.3, our interests fully align in this case. When it comes to continuity, the authors keep the same model by perceiving discrete spaces as infinitesimal motions, whereas our continuous machines live as spans inside a completely different topos—sheaves on real intervals.

Finally, our notion of machines was motivated by a project with NASA and Honeywell, in which we were tasked with giving semantics to a system of interacting systems, such as the US national airspace system. This work later morphed into [SS17] which emphasizes the logical aspects, whereas the current paper is geared toward an understanding of control-theoretic issues: when do systems of systems, with components of different types, still exhibit deterministic control based on inputs. The work is ongoing, e.g. we hope to understand the semantics of probabilistic structures (internal valuations), so that we can model probabilistic behaviors, such as stochastic processes.

Plan of the paper. In Chapter 2, we briefly review monoidal categories and (colored) operads as well as the theory of their algebras, and we describe the category $\mathcal{W}_\mathcal{C}$ of labeled boxes and wiring diagrams for a typing category $\mathcal{C}$. We recall the $\mathcal{W}_\mathcal{C}$-algebras of discrete

and continuous dynamical systems, for which wires carry only static information—the set of symbols or the space of parameters that drive the system—and finally we introduce the general $\mathcal{C}$-span algebras.

We bring time into the picture in Chapter 3, where we define sites **Int** and **Int**$_N$ of continuous and discrete-time intervals respectively, and consider their toposes of sheaves. We also provide morphisms from discrete and continuous sheaves to a slice topos $\widetilde{\textbf{Int}}/\text{Sync}$, where Sync is a *synchronizing* sheaf whose sections can be thought of as phase shifts.

We define abstract dynamical systems, which we call *machines*, in Chapter 4. They form $\mathcal{W}_{\mathcal{C}}$-algebras, and so do various subclasses such as machines that are total and/or deterministic. For $\mathcal{C} = \widetilde{\textbf{Int}}, \widetilde{\textbf{Int}}_N$ or $\widetilde{\textbf{Int}}/\text{Sync}$ we obtain different time-typed machines, designed to encompass as general systems as possible, and we also discuss a notion of *safety contracts* for machines.

In Chapter 5 we provide algebra morphisms that translate between almost every combination of machines discussed previously. In particular, we convert our older, static definitions of discrete and continuous dynamical systems into our new language of time-based machines, and also discrete and continuous machines into synchronous machines. We give sufficient conditions for these translations to preserve totality and determinism.

Finally, in Appendix A we give an alternative categorical construction of the sheaf topos $\widetilde{\textbf{Int}}$, namely as the category of *discrete Conduché fibrations* over the monoid of nonnegative real numbers. This equivalence is derived from results in [Joh99] and provides another viewpoint for machines and their variations.

2. Wiring diagrams and algebras

This paper is about interconnecting systems in order to build more complex systems. The notion of building one object from many is nicely captured using operads and their algebras: an operad describes the ways 'objects' can be combined, and an operad algebra gives semantics by describing the 'objects' themselves. The operads used in this article in fact all underlie monoidal categories. While we assume familiarity with monoidal categories, we review some key facts and the relationship to operads in Section 2.1. Standard references for these topics include [JS93] and [Lei04a]. The notion of a multicategory (colored operad) was in fact introduced much earlier by Lambek [Lam69], along with the crucial observation that every monoidal category gives rise to such a structure.

In Section 2.2 we restrict our interest to a certain symmetric monoidal category $\mathcal{W}$ (and its associated operad) studied in previous works like [VSL15]. The objects and morphisms in $\mathcal{W}$ classify a certain sort of string diagrams, which for reasons explained elsewhere [SSR16] we call wiring diagrams. The category of wiring diagram algebras, i.e. monoidal functors $\mathcal{W} \to \textbf{Cat}$, is the realm where various notions of systems live and can be composed

according to their wiring arrangements. In Section 2.3 we elaborate on two important examples of those algebras, namely discrete and continuous dynamical systems, whereas in Section 2.4 we describe a very general class of span-like algebras, whose semantics is fundamental for our description of abstract systems in Chapter 4.

2.1. **Background on monoidal categories and operads.** We denote a monoidal category by $(\mathcal{V}, \otimes, I)$. Recall that a *lax monoidal* functor $F\colon (\mathcal{V}, \otimes, I) \to (\mathcal{W}, \otimes, I)$ comes equipped with natural structure morphisms

$$F_{c,d}\colon F(c) \otimes F(d) \to F(c \otimes d) \quad \text{and} \quad F_I\colon I \to F(I)$$

satisfying well-known coherence axioms. The functor F is *strong monoidal* if these structure morphisms are isomorphisms. If the monoidal categories $\mathcal{V}$ and $\mathcal{W}$ are moreover symmetric, with $b_{c,d}\colon c \otimes d \xrightarrow{\sim} d \otimes c$, then F is *symmetric* if the structure maps appropriately commute with the symmetry isomorphisms. A *monoidal* natural transformation $\alpha\colon F \Rightarrow G$ between monoidal functors is an ordinary natural transformation whose components furthermore commute with the structure morphisms. For detailed descriptions, see e.g. [JS93].

We denote by $\mathbf{SMC}_\ell$ the 2-category of symmetric monoidal categories, symmetric lax monoidal functors and monoidal natural transformations, and by $\mathbf{SMC}$ the corresponding 2-category with strong monoidal functors.

Two fundamental examples of cartesian monoidal categories are $(\mathbf{Set}, \times, \{*\})$ and $(\mathbf{Cat}, \times, \mathbf{1})$. In our context, we will often refer to symmetric lax monoidal functors $\mathcal{V} \to \mathbf{Cat}$ as ($\mathbf{Cat}$-valued) $\mathcal{V}$-*algebras*, and to monoidal natural transformations between them as $\mathcal{V}$-*algebra maps*; hence we denote $\mathcal{V}\text{-}\mathbf{Alg} \coloneqq \mathbf{SMC}_\ell(\mathcal{V}, \mathbf{Cat})$. The terminology comes from the closely-connected world of operads, to which we next turn.

Recall that a *colored operad*, or *multicategory*, $\mathcal{P}$ consists of a set of objects (colors) ob $\mathcal{P}$, a hom-set of n-ary operations $\mathcal{P}(c_1, \ldots, c_n; c)$ for each $(n+1)$-tuple of objects, an identity operation $\mathrm{id}_c \in \mathcal{P}(c; c)$, and a composition formula

$$\mathcal{P}(c_1, \ldots, c_n; c) \times \mathcal{P}(c_{11}, \ldots, c_{1k_1}; c_1) \times \cdots \times \mathcal{P}(c_{n1}, \ldots, c_{nk_n}; c_n) \to \mathcal{P}(c_{11}, \ldots, c_{nk_n}; c)$$

that can be visualized [Lei04a, Fig. 2-B] as

(1)
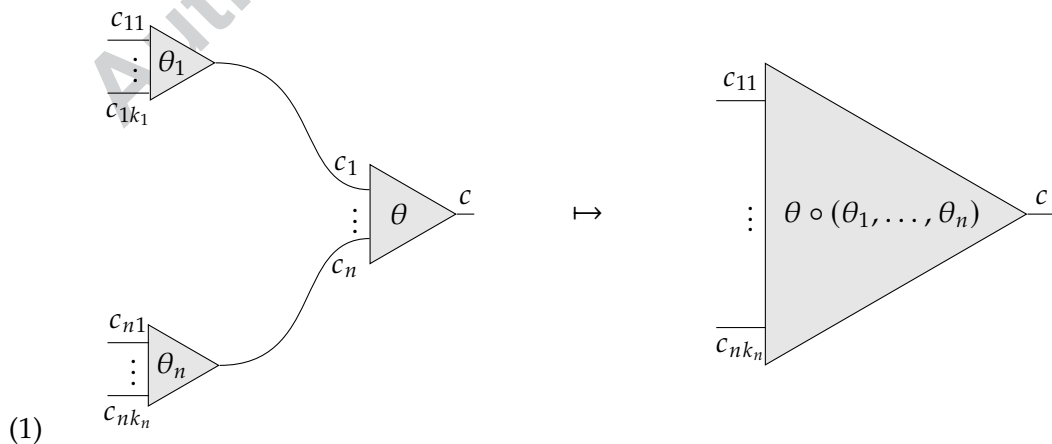
This data is subject to associativity and unitality axioms, and the operad is moreover *symmetric* if there are compatible permutation actions on the hom-sets, $\mathcal{P}(c_1, \ldots, c_n; c) \xrightarrow{\sim} \mathcal{P}(c_{\sigma(1)}, \ldots, c_{\sigma(n)}; c)$. An *operad functor* $F\colon \mathcal{P} \to \mathcal{P}'$ consists of a mapping on objects, and a

mapping on hom-sets $\mathcal{P}(c_1, \ldots, c_n; c) \to \mathcal{P}'(Fc_1, \ldots, Fc_n; Fc)$ that preserves composition, symmetries and identities. An *operad transformation* $\alpha\colon F \Rightarrow G$ consists of unary operations $\alpha_c \in \mathcal{P}'(Fc; Gc)$ that are compatible with respect to the composition formula. Therefore we obtain a 2-category $\mathbf{SOpd}$, often denoted $\mathbf{SMulticat}$ in the literature, of symmetric colored operads; see e.g. [Her00, Lei04a].

There exists a 2-functor, called the *underlying operad* functor as in [Lei04a, Example 2.1.3],

$$(2) \qquad\qquad \mathcal{O}\colon \mathbf{SMC}_\ell \to \mathbf{SOpd}$$

mapping each symmetric monoidal category $\mathcal{V}$ to the operad with $\mathrm{ob}(\mathcal{OV}) \coloneqq \mathrm{ob}\,\mathcal{V}$ and $\mathcal{OV}(c_1, \ldots, c_n; c) \coloneqq \mathcal{V}(c_1 \otimes \cdots \otimes c_n, c)$. Note that for $c = c_1 \otimes \cdots \otimes c_n$, there is a unique morphism in $\mathcal{OV}$ corresponding to id_c, called the *universal morphism* for $(c_1, \ldots, c_n)$. To each lax monoidal functor $F\colon \mathcal{V} \to \mathcal{W}$, we can assign an operad functor $\mathcal{O}F$ with the same mapping on objects. On morphisms, it sends some $f\colon c_1 \otimes \cdots \otimes c_n \to c$ in $\mathcal{V}$ to the composite

$$Fc_1 \otimes \cdots \otimes Fc_n \xrightarrow{F_{c_1, \cdots, c_n}} F(c_1 \otimes \cdots \otimes c_n) \xrightarrow{Ff} Fc$$

in $\mathcal{W}$. Lastly, a monoidal natural transformation α involves exactly the data needed to define the operad transformation $\mathcal{O}\alpha$.

The functor $\mathcal{O}$ is clearly faithful, but it is also full: for any operad map $G\colon \mathcal{OV} \to \mathcal{OW}$, we can define a functor $F\colon \mathcal{V} \to \mathcal{W}$ via the action of G on objects and unary morphisms. Its lax monoidal structure is given by the action of G on universal morphisms. Moreover, there is also a bijection between the respective 2-cells, which exhibits $\mathcal{O}$ as a fully faithful 2-functor; see [Her00, Remark 9.5] for relevant results. Not all operads underlie monoidal categories—the functor $\mathcal{O}$ is not essentially surjective—however, all operads of interest in this paper do happen to be in the image of $\mathcal{O}$.

For typographical reasons, we denote the operad $\mathcal{O}\mathbf{Cat}$ underlying the cartesian monoidal category of categories using the same symbol $\mathbf{Cat}$, when no confusion arises. If $\mathcal{P}$ is an operad, a ($\mathbf{Cat}$-valued) $\mathcal{P}$-*algebra* is an operad functor $\mathcal{P} \to \mathbf{Cat}$. The category of $\mathcal{P}$-algebras is denoted $\mathcal{P}\text{-}\mathbf{Alg} \coloneqq \mathbf{SOpd}(\mathcal{P}, \mathbf{Cat})$, see [Lei04a, Def 2.1.12]. This nomenclature agrees with that above, since for any monoidal category $\mathcal{V}$, the fully faithful $\mathcal{O}$ induces an isomorphism between the corresponding categories of algebras

$$(3) \qquad\qquad \mathcal{V}\text{-}\mathbf{Alg} \xrightarrow{\sim} (\mathcal{OV})\text{-}\mathbf{Alg}.$$

Remark 2.1.1. In this article, we will in fact work with $\mathcal{W}$-*pseudoalgebras* for various monoidal categories $\mathcal{W}$, rather than ordinary algebras. This amounts to viewing $\mathcal{W}$ as a monoidal 2-category (with trivial 2-cells), and considering *weak monoidal pseudofunctors* $A\colon (\mathcal{W}, \otimes, I) \to (\mathbf{Cat}, \times, \mathbf{1})$. More explicitly, for any pair of maps ψ, ϕ in $\mathcal{W}$, we have only a natural isomorphism of functors $A(\psi \circ \phi) \cong A(\psi) \circ A(\phi)$ rather than a strict equality; similarly for the identities, along with coherence axioms. Moreover, the lax monoidal structure is pseudo, meaning that any previously commutative diagram of the axioms now only commutes up to natural isomorphism (with certain coherence conditions tying them together). In fact, $\mathcal{W}$-algebras are really *monoidal indexed categories* in the sense of [MV18].

More details of such structures can be found e.g. in [DS97], where $\mathbf{WMonHom}(\mathcal{A}, \mathcal{B})$ denotes the category of weak monoidal pseudofunctors and monoidal pseudonatural

transformations, for any two monoidal 2-categories $\mathcal{A}, \mathcal{B}$. Due to coherence theorems, there is an equivalence $\mathcal{W}\text{-}\mathbf{Alg} \simeq \mathbf{WMonHom}(\mathcal{W}, \mathbf{Cat})$ and so we safely omit such details in the presentation below, to keep the already-technical material more readable. Thus we will speak of the pseudoalgebra A as above simply as a $\mathcal{W}$-algebra and the pseudomaps between them simply as $\mathcal{W}$-algebra morphisms.

2.2. **The operad of labeled boxes and wiring diagrams.** Operads (or monoidal categories) of various 'wiring diagram shapes' have been considered in [Spi13, RS13]. More recently, [SSR16] showed a strong relationship between the category of algebras on a certain operad (namely **Cob**, the operad of oriented 1-dimensional cobordisms) and the category of traced monoidal categories; results along similar lines were proven in [Fon16]. The operads $\mathcal{W}$ we use here are designed to model what could be called 'cartesian traced categories without identities'.[1]

For any category $\mathcal{C}$, a $\mathcal{C}$-*typed finite set* is a finite set X together with a function $\tau \colon X \to \mathrm{ob}\,\mathcal{C}$ assigning to every element $x \in X$ an object $\tau(x) \in \mathcal{C}$, called its *type*. We often elide the typing function in notation when it is implied from the context, writing X rather than (X, τ). The $\mathcal{C}$-typed finite sets form a category $\mathbf{TFS}_{\mathcal{C}}$, where morphisms are functions $f \colon X \to X'$ that respect the types, i.e. $\tau(fx) = \tau(x)$; we call such an f a $\mathcal{C}$-*typed function*.[2] Thus we have

$$\mathbf{TFS}_{\mathcal{C}} \cong \mathbf{FinSet}/\mathrm{ob}\,\mathcal{C}$$

and $\mathcal{C}$ is called the *typing* category. Notice that $(\mathbf{TFS}_{\mathcal{C}}, +, \emptyset)$ is a cocartesian monoidal category via

$$(X, \tau) + (X', \tau') = \big(X + X', (\tau, \tau')\big)$$

where $X + X'$ is the disjoint union of X, X' and (τ, τ') denotes the co-pairing of functions. Moreover, any functor $F \colon \mathcal{C} \to \mathcal{D}$ induces a (symmetric strong monoidal) functor $\mathbf{TFS}_F \colon \mathbf{TFS}_{\mathcal{C}} \to \mathbf{TFS}_{\mathcal{D}}$, sending (X, τ) to $(X, F \circ \tau)$. We have thus constructed a functor $\mathbf{TFS}_{(-)} \colon \mathbf{Cat} \to \mathcal{SMC}$.

If $\mathcal{C}$ has finite products, then we can assign to each $\mathcal{C}$-typed finite set (X, τ) the product of all types of its elements, denoted $\widehat{X} \in \mathcal{C}$:

$$(4) \qquad\qquad \widehat{X} := \prod_{x \in X} \tau(x)$$

This induces a functor $\widehat{(-)}_{\mathcal{C}} \colon \mathbf{TFS}_{\mathcal{C}}^{\mathrm{op}} \to \mathcal{C}$, which is strong monoidal because there are isomorphisms $\widehat{X_1} \times \widehat{X_2} \cong \widehat{X_1 + X_2}$ and $\widehat{\emptyset} \cong 1$. In fact, if $\mathbf{FPCat}$ is the category of finite-product categories and functors which preserve them, and $i \colon \mathbf{FPCat} \to \mathcal{SMC}$ realizes a

---

[1]The reason to study traced categories without identities is that there is no 'identity' dynamical system. Attempting to define such a thing, one loses the trace structure; similar problems arise throughout computer science and engineering applications: see [Age96, Sel97]. A solution to this problem was described in [ABP99], using trace ideals. The present operadic approach is another solution, which appears to be roughly equivalent.

[2] The category $\mathbf{TFS}_{\mathcal{C}}$ of typed finite sets in $\mathcal{C}$ is closely related to the *finite family fibration* $\mathbf{Fam}(\mathcal{C}^{\mathrm{op}}) \to \mathbf{FinSet}$ (see [Jac99, Definition 1.2.1]); namely $\mathbf{TFS}_{\mathcal{C}}$ is the category of cartesian arrows in $\mathbf{Fam}(\mathcal{C}^{\mathrm{op}})$, or equivalently cartesian arrows in $\mathbf{Fam}(\mathcal{C})$. Replacing $\mathbf{TFS}_{\mathcal{C}}$ with $\mathbf{Fam}(\mathcal{C}^{\mathrm{op}})$ throughout the paper would yield similar definitions and results; for example, taking product of types (4) is well-defined in this context, $\prod \colon \mathbf{Fam}(\mathcal{C}^{\mathrm{op}})^{\mathrm{op}} \to \mathcal{C}$.

finite-product category with its canonical cartesian monoidal structure, then the functors $\widehat{(-)}_{\mathcal{C}}$ are the components of a pseudonatural transformation

(5)

$$
\begin{array}{ccc}
 & \mathbf{Cat} & \xrightarrow{\ \mathbf{TFS}^{\mathrm{op}}_{(-)}\ } \\
\nearrow & & \searrow \\
\mathbf{FPCat} & \Downarrow\widehat{(-)} & \mathbf{SMC} \\
& \searrow_{\ i\ } \nearrow &
\end{array}
$$

where the ordinary category **FPCat** and functor **TFS** are viewed as 2-structures with the trivial 2-cells and mappings respectively.

Example 2.2.1. Typed finite sets will be heavily used in the context of wiring diagrams; here we consider a few typing categories $\mathcal{C}$, whose objects will serve as types of elements later. In general, if (X, τ) is a typed finite set, we think of each element $x \in X$ as a *port*, while its associated type $\tau(x) \in \mathrm{ob}\,\mathcal{C}$ specifies what sort of information passes through that port.

  (1) When $\mathcal{C} = \mathbf{Set}$, each port carries a *set* of possible signals; there is no notion of time. Ports of this type are relevant when studying discrete dynamical systems.
  (2) When $\mathcal{C} = \mathbf{Man}$, (a small category equivalent to) the category of second-countable smooth manifolds and smooth maps between them, each port carries a manifold of possible signals, again without a notion of time. Ports of this type are relevant when specifying continuous dynamical systems: machines that behave according to (ordinary) differential equations on manifolds.
  (3) When $\mathcal{C} = \widetilde{\mathbf{Int}}$, the category of *interval sheaves* defined in Section 3.1, each port carries a very general kind of time-based signal. The two types of dynamical systems described above in terms of sets and manifolds can be translated into this language, at which point the time-based dynamics itself becomes evident. The topos of interval sheaves will be central in our construction of machines.

We are now in position to define a symmetric monoidal category $(\mathcal{W}_{\mathcal{C}}, \oplus, 0)$ for any typing category $\mathcal{C}$, see [VSL15, §3] or [Spi15, §3.3]. Its objects will be called $\mathcal{C}$-*labeled boxes*; they are pairs $X = (X^{\mathrm{in}}, X^{\mathrm{out}}) \in \mathbf{TFS}_{\mathcal{C}} \times \mathbf{TFS}_{\mathcal{C}}$ of $\mathcal{C}$-typed finite sets. We can picture such an X as

(6)

Here, $X^{\mathrm{in}} = \{a_1, \dots, a_m\}$ is the set of input ports, and $X^{\mathrm{out}} = \{b_1, \dots, b_n\}$ is the set of output ports. Each port a_i or b_j comes with its associated $\mathcal{C}$-type $\tau(a_i)$ or $\tau(b_j) \in \mathcal{C}$.

A morphism $\phi\colon X \to Y$ in $\mathcal{W}_{\mathcal{C}}$ is called a *wiring diagram*. It consists of a pair

(7)
$$
\begin{cases}
\phi^{\mathrm{in}}\colon X^{\mathrm{in}} \to X^{\mathrm{out}} + Y^{\mathrm{in}} \\
\phi^{\mathrm{out}}\colon Y^{\mathrm{out}} \to X^{\mathrm{out}}
\end{cases}
$$

of $\mathcal{C}$-typed functions which express 'which port is fed information by which'. Graphically, we can picture such a map ϕ, going from the inside box X to the outside box Y, as

(8)
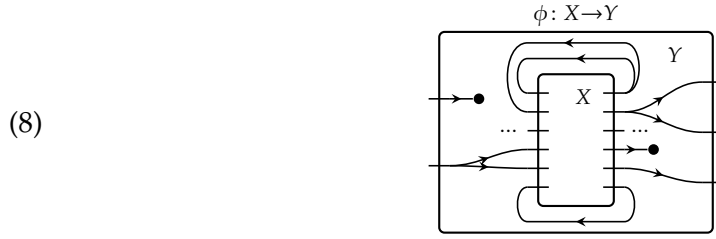
The identity wiring diagram $1_X = (1_X^{\text{in}}\colon X^{\text{in}} \to X^{\text{out}} + X^{\text{in}}, 1_X^{\text{out}}\colon X^{\text{out}} \to X^{\text{out}})$ is given by the coproduct inclusion and identity respectively. Given another wiring diagram $\psi = (\psi^{\text{in}}, \psi^{\text{out}})\colon Y \to Z$, their composite $\psi \circ \phi =: \omega = (\omega^{\text{in}}, \omega^{\text{out}})$ is given by

$$\omega^{\text{in}}\colon X^{\text{in}} \xrightarrow{\phi^{\text{in}}} X^{\text{out}}+Y^{\text{in}} \xrightarrow{1+\psi^{\text{in}}} X^{\text{out}}+Y^{\text{out}}+Z^{\text{in}} \xrightarrow{1+\phi^{\text{out}}+1} X^{\text{out}}+X^{\text{out}}+Z^{\text{in}} \xrightarrow{\nabla+1} X^{\text{out}}+Z^{\text{in}}$$

$$\omega^{\text{out}}\colon Z^{\text{out}} \xrightarrow{\psi^{\text{out}}} Y^{\text{out}} \xrightarrow{\phi^{\text{out}}} X^{\text{out}}$$

Associativity and unitality can be verified; see [Spi15, Def 3.10] for details.

There is a monoidal structure on $\mathcal{W}_{\mathcal{C}}$ as follows. For two labeled boxes $X_1 = (X_1{}^{\text{in}}, X_1{}^{\text{out}})$ and $X_2 = (X_2{}^{\text{in}}, X_2{}^{\text{out}})$, their tensor product $X_1 \oplus X_2$ is defined to be $(X_1{}^{\text{in}} + X_2{}^{\text{in}}, X_1{}^{\text{out}} + X_2{}^{\text{out}})$ and amounts to parallel composition
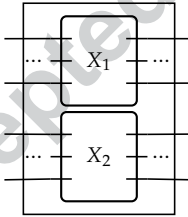
of boxes, viewed as a new box. The monoidal unit is $0 = (\emptyset, \emptyset)$. We refer to $(\mathcal{W}_{\mathcal{C}}, \oplus, 0)$ as the symmetric monoidal category of $\mathcal{C}$-*labeled boxes and wiring diagrams*

An arbitrary functor $F\colon \mathcal{C} \to \mathcal{D}$ induces a symmetric strong monoidal functor $\mathcal{W}_F\colon \mathcal{W}_{\mathcal{C}} \to \mathcal{W}_{\mathcal{D}}$. It maps objects $X = ((X^{\text{in}}, \tau^{\text{in}}), (X^{\text{out}}, \tau^{\text{out}}))$ to $\mathcal{W}_F(X) = ((X^{\text{in}}, F \circ \tau^{\text{in}}), (X^{\text{out}}, F \circ \tau^{\text{out}}))$ which consist of the same finite set of input and output ports, but with new typing functions defined by F. Together with the identity-like action of $\mathcal{W}_F$ on morphisms, we obtain a functor

(9) $$\mathcal{W}_{(-)}\colon \mathbf{Cat} \to \mathcal{SMC}.$$

Even though this is defined on arbitrary categories and functors, it turns out that in many of the examples that follow, using products (4) and pullbacks is essential; see also Remark 2.2.2. Thus we often restrict to finitely complete categories $\mathbf{FCCat} \rightarrowtail \mathbf{Cat}$ as the domain of $\mathcal{W}_{(-)}$, e.g. see Proposition 2.4.4.

Following Example 2.2.1, some wiring diagram categories we will use include $\mathcal{W}_{\mathbf{Set}}$, $\mathcal{W}_{\mathbf{Euc}}$ and $\mathcal{W}_{\widetilde{\mathbf{Int}}}$. Regarding induced functors between them, consider the example of the limit-preserving $U\colon \mathbf{Euc} \to \mathbf{Set}$ which maps a Euclidean space to its underlying set. This naturally induces a symmetric strong monoidal functor

(10) $$\mathcal{W}_U\colon \mathcal{W}_{\mathbf{Euc}} \to \mathcal{W}_{\mathbf{Set}}$$

between the respective categories of wiring diagrams, important for Proposition 2.3.5.

If we apply the underlying operad functor (2) to any monoidal category $\mathcal{W}_\mathcal{C}$, we obtain the *operad of wiring diagrams* $\mathcal{O}\mathcal{W}_\mathcal{C}$. An object, or color, is a $\mathcal{C}$-labeled box (6), whereas for example a 5-ary morphism $\phi\colon X_1, \ldots, X_5 \to Y$ in $\mathcal{O}\mathcal{W}_\mathcal{C}$ can be drawn like

(11)
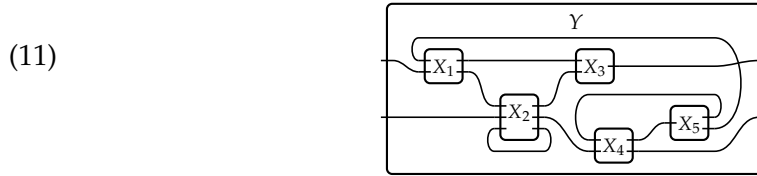
Notice that the boxes are the objects and not the morphisms in the underlying operad $\mathcal{O}\mathcal{W}_\mathcal{C}$, as the earlier triangles representation for n-ary operad morphisms might suggest; therefore operadic composition as in (1) in this framework corresponds to a zoomed-in picture of a box three layers 'deep', as in Fig. 1.

In what follows, our goal is to model various processes as objects of $(\mathcal{O}\mathcal{W}_\mathcal{C})$-**Alg**, i.e. algebras for this operad. Due to the isomorphism (3) between algebras for a monoidal category and for its underlying operad, we can identify such an operad algebra with a lax monoidal functor from $(\mathcal{W}_\mathcal{C}, +, 0)$ to $(\textbf{Cat}, \times, \mathbf{1})$, namely a $\mathcal{W}_\mathcal{C}$-algebra. Given such an algebra $F\colon \mathcal{W}_\mathcal{C} \to \textbf{Cat}$ and a $\mathcal{C}$-labeled box X (6), we refer to the objects of the category $F(X)$ as *inhabitants* of X. An algebra provides semantics to the boxes, examples of which we will encounter at Section 2.3 as well as Chapter 4. Whereas the formal description of the algebras will be completely in terms of lax monoidal functors, having an associated operadic interpretation provides meaningful pictorial representations like (11).

In more detail, given inhabitants of each inside box and their arrangement ϕ, an algebra F constructs an inhabitant of the outside box, thus associating a sort of *composition formula* to ϕ. Indeed, for any algebra $F\colon \mathcal{W}_\mathcal{C} \to \textbf{Cat}$, the composite functor

(12) $$F(X_1) \times \cdots \times F(X_5) \xrightarrow{F_{X_1,\ldots,X_5}} F(X_1 + \cdots + X_5) \xrightarrow{F(\phi)} F(Y)$$

performs the following two steps. The F-inhabitants of five boxes $X_1, \ldots, X_5$ are first combined—using the lax structure morphisms of F—to form a single inhabitant of the parallel composite box $X := X_1 + \cdots + X_5$. The wiring diagram ϕ can then be considered 1-ary, and the functor $F(\phi)$ converts the inhabitant of X to an inhabitant of the outer box Y.

Finally, there exists a category with objects symmetric lax monoidal functors $F\colon \mathcal{W}_\mathcal{C} \to \textbf{Cat}$[3] with domains categories of $\mathcal{C}$-labeled boxes and wiring diagrams, and morphisms

(13)

$$
\begin{array}{ccc}
\mathcal{W}_\mathcal{C} & \xrightarrow{F} & \\
\mathcal{W}_F \downarrow & \Downarrow \alpha & \textbf{Cat} \\
\mathcal{W}_\mathcal{D} & \xrightarrow{G} &
\end{array}
$$

where $\mathcal{W}_F$ is the symmetric strong monoidal functor induced by some functor $F\colon \mathcal{C} \to \mathcal{D}$ as in (9), and α is a monoidal natural transformation. Denote this category by **WD-Alg**[4]

---

[3]In fact, F should be a symmetric weak monoidal pseudofunctor and $\alpha\colon F \to GW_F$ should be a monoidal pseudonatural transformation; we sweep these details under the rug as discussed in Remark 2.1.1.

[4]This is a subcategory of **MonOpICat** of the tensor objects in (op)indexed categories, see [MV18].

and notice there is an obvious forgetful functor $\mathbf{WD\text{-}Alg} \xrightarrow{\text{dom}} \mathcal{SMC}$. All of our algebras examples like discrete and continuous dynamical systems from Section 2.3, $\mathsf{Spn}_{\mathcal{C}}$ from Proposition 2.4.1 and machines from Chapter 4 are objects of $\mathbf{WD\text{-}Alg}$, whereas Chapter 5 describes various maps between them, which allow the translation of one system kind to another.

Remark 2.2.2. Suppose the typing category $\mathcal{C}$ has finite products and consider a $\mathcal{C}$-labeled box $X = (X^{\text{in}}, X^{\text{out}})$ as in (6). By forming the products $\widehat{X^{\text{in}}} = \prod_{x \in X^{\text{in}}} \tau(x)$ and $\widehat{X^{\text{out}}} = \prod_{x \in X^{\text{out}}} \tau(x)$, we can associate to the entire input side (all its ports) a single object $\widehat{X^{\text{in}}} \in \mathcal{C}$, and similarly associate $\widehat{X^{\text{out}}} \in \mathcal{C}$ to the entire output side. Since the functor $\widehat{(\,\cdot\,)}_{\mathcal{C}}$ (5) is contravariant and strong monoidal, a wiring diagram $\phi = (\phi^{\text{in}}, \phi^{\text{out}})$ (7) induces a pair of morphisms in $\mathcal{C}$

$$(14) \qquad \begin{cases} \widehat{\phi^{\text{in}}} \colon \widehat{Y^{\text{in}}} \times \widehat{X^{\text{out}}} \to \widehat{X^{\text{in}}} \\ \widehat{\phi^{\text{out}}} \colon \widehat{X^{\text{out}}} \to \widehat{Y^{\text{out}}} \end{cases}$$

Intuitively, these now describe the direction of information flow in the wiring diagram: in (8) the box X receives information only from the input of Y as well as from X's output (feedback operation), whereas information that exits Y only came from the output of X.

Through all examples of $\mathcal{W}_{\mathcal{C}}$-algebras $F \colon \mathcal{W}_{\mathcal{C}} \to \mathbf{Cat}$ found in this paper, a pattern arises: $\mathcal{C}$ will be a finitely complete category, and we always begin by taking the product of types. That is, all of our example algebras F factor as

$$\begin{array}{ccc} \mathcal{W}_{\mathcal{C}} & \xrightarrow{\quad F \quad} & \mathbf{Cat} \\ {\scriptstyle X \mapsto \widehat{X}} \searrow & & \nearrow {\scriptstyle \overline{F}} \\ & \overline{\mathcal{W}}_{\mathcal{C}} & \end{array}$$

where $\overline{\mathcal{W}}_{\mathcal{C}}$ is the monoidal category with objects $\mathrm{ob}(\mathcal{C} \times \mathcal{C})$, and morphisms $(X_1, X_2) \to (Y_1, Y_2)$ pairs of $\mathcal{C}$-morphisms

$$\begin{cases} \phi_1 \colon Y_1 \times X_2 \to X_1 \\ \phi_2 \colon X_2 \to Y_2 \end{cases}$$

with appropriate composition, identities and monoidal structure. The map $\mathcal{W}_{\mathcal{C}} \to \overline{\mathcal{W}}_{\mathcal{C}}$ sends $X = (X^{\text{in}}, X^{\text{out}}) \mapsto (\widehat{X^{\text{in}}}, \widehat{X^{\text{out}}})$. We will not mention $\overline{\mathcal{W}}_{\mathcal{C}}$ again, though we often slightly abuse notation by writing F in place of $\overline{F}$.

The above description of $\overline{\mathcal{W}}_{\mathcal{C}}$ is reminiscent of compositional game theory [Hed18] and there also seem to be connections with bilenses and the Dialectica category [Pai90]; such considerations are in the center of future research goals.

2.3. Discrete and continuous dynamical systems.

As our primary examples, we consider two well-known classes of dynamical systems, namely *discrete* and *continuous*, which have already been studied within the context of the operad of wiring diagrams in previous works; see e.g. [RS13, Spi15]. A broad goal of this paper is to group these, as well as other notions of systems, inside a generalized framework. This will be accomplished by constructing algebra maps from these special case systems to the new, abstracted ones, described in Chapter 5.

Definition 2.3.1. A *discrete dynamical system* with *input set A* and *output set B* consists of a set S, called the *state set*, together with two functions

$$\begin{cases} f^{\mathrm{upd}} \colon A \times S \to S \\ f^{\mathrm{rdt}} \colon S \to B \end{cases}$$

respectively called *update* and *readout functions*, which express the transition operation and the produced output of the machine. We refer to $(S, f^{\mathrm{upd}}, f^{\mathrm{rdt}})$ as an (A, B)-*discrete dynamical system* or (A, B)-*DDS*. If additionally an element $s_0 \in S$ is chosen, we refer to $(S, s_0, f^{\mathrm{upd}}, f^{\mathrm{rdt}})$ as an *initialized* (A, B)-*DDS*.[5]

A *morphism* $(S_1, f_1^{\mathrm{upd}}, f_1^{\mathrm{rdt}}) \to (S_2, f_2^{\mathrm{upd}}, f_2^{\mathrm{rdt}})$ of (A, B)-discrete dynamical systems is a function $h \colon S_1 \to S_2$ which commutes with the update and readout functions elementwise, i.e. $h(f_1^{\mathrm{upd}}(s, x)) = f_2^{\mathrm{upd}}(hs, x)$ and $f_1^{\mathrm{rdt}}(s) = f_2^{\mathrm{rdt}}(hs)$.

The resulting category is denoted by $\mathsf{DDS}(A, B)$. Notice the similarity of the above description with a morphism inside $\overline{\mathcal{W}}$ of Remark 2.2.2: a discrete dynamical system is a $\overline{\mathcal{W}}_{\mathbf{Set}}$-map $(S, S) \to (A, B)$ in that sense.

We can now define a functor $\mathsf{DDS} \colon \mathcal{W}_{\mathbf{Set}} \to \mathbf{Cat}$, as in [Spi15, Def. 4.1,4.6,4.11]. To an object $X = (X^{\mathrm{in}}, X^{\mathrm{out}}) \in \mathcal{W}_{\mathbf{Set}}$ we assign the category $\mathsf{DDS}(X) := \mathsf{DDS}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$ as defined above. To a morphism (wiring diagram) $\phi \colon X \to Y$ (14), we define $\mathsf{DDS}(\phi) \colon \mathsf{DDS}(X) \to \mathsf{DDS}(Y)$ to be the functor which sends the system $(S, f^{\mathrm{upd}}, f^{\mathrm{rdt}}) \in \mathsf{DDS}(X)$ to the system $(S, g^{\mathrm{upd}}, g^{\mathrm{rdt}}) \in \mathsf{DDS}(Y)$ having the same state set S, and new update and readout functions given by

$$g^{\mathrm{upd}} \colon \widehat{Y^{\mathrm{in}}} \times S \xrightarrow{1 \times \Delta} \widehat{Y^{\mathrm{in}}} \times S \times S \xrightarrow{1 \times f^{\mathrm{rdt}} \times 1} \widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \times S \xrightarrow{\widehat{\phi^{\mathrm{in}}} \times 1} \widehat{X^{\mathrm{in}}} \times S \xrightarrow{f^{\mathrm{upd}}} S$$

$$g^{\mathrm{rdt}} \colon S \xrightarrow{f^{\mathrm{rdt}}} \widehat{X^{\mathrm{out}}} \xrightarrow{\widehat{\phi^{\mathrm{out}}}} \widehat{Y^{\mathrm{out}}}$$

also written, via their mappings on elements, as

$$(15) \qquad g^{\mathrm{upd}}(y, s) := f^{\mathrm{upd}}\left(\widehat{\phi^{\mathrm{in}}}(y, f^{\mathrm{rdt}}(s)), s\right) \quad \text{and} \quad g^{\mathrm{rdt}}(s) := \widehat{\phi^{\mathrm{out}}}(f^{\mathrm{rdt}}(s))$$

It can be verified that $\mathsf{DDS}(\phi)$ preserves composition and identities in $\mathcal{W}_{\mathbf{Set}}$. Moreover, DDS has a symmetric lax monoidal structure essentially given by cartesian product: for systems $F = (S, f^{\mathrm{upd}}, f^{\mathrm{rdt}}) \in \mathsf{DDS}(X)$ and $G = (T, g^{\mathrm{upd}}, g^{\mathrm{rdt}}) \in \mathsf{DDS}(Y)$, we define the functor $\mathsf{DDS}_{X,Y} \colon \mathsf{DDS}(X) \times \mathsf{DDS}(Y) \to \mathsf{DDS}(X + Y)$ to map (F, G) to

$$(S \times T, \widehat{X^{\mathrm{in}}} \times \widehat{Y^{\mathrm{in}}} \times S \times T \xrightarrow{\sim} \widehat{X^{\mathrm{in}}} \times S \times \widehat{Y^{\mathrm{in}}} \times T \xrightarrow{f^{\mathrm{upd}} \times g^{\mathrm{upd}}} S \times T, S \times T \xrightarrow{f^{\mathrm{rdt}} \times g^{\mathrm{rdt}}} \widehat{X^{\mathrm{out}}} \times \widehat{Y^{\mathrm{out}}}).$$

Proposition 2.3.2. *There exists a $\mathcal{W}_{\mathbf{Set}}$-algebra DDS, for which the labeled boxes inhabitants are discrete dynamical systems.*

In particular, if we have any interconnection arrangement of discrete dynamical systems like (11), we can use this algebra structure to explicitly construct the state set, the update and the readout function of the new discrete dynamical system which the subsystems form.

---

[5] Note that if A, B, S are finite, what we have called an initialized (A, B)-DDS is often called a *Moore machine*.

Notice that the minor abuse of notation $\mathsf{DDS}(X) := \mathsf{DDS}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$ is explained by Remark 2.2.2. The algebra DDS is a prototype of many algebras throughout this paper, e.g. the abstract systems in Chapter 4. One will see strong similarities in what we describe next, the algebra of continuous dynamical systems (details on which can be found in [VSL15, §4] or [Spi15, §2.4]).

Definition 2.3.3. Let A, B be Euclidean spaces. An (A, B)-*continuous dynamical system* is a Euclidean space S, called the *state space*, equipped with smooth functions

$$\begin{cases} f^{\mathrm{dyn}} \colon A \times S \to TS \\ f^{\mathrm{rdt}} \colon S \to B \end{cases}$$

where TS is the tangent bundle of S, such that f^{dyn} commutes with the projections to S. That is, $f^{\mathrm{dyn}}(a, s) = (s, v)$ for some vector v; it is standard notation to write $(f^{\mathrm{dyn}})_{a,s} := v$. The maps f^{dyn} and f^{rdt} are respectively called the *dynamics* and *readout* functions. The first is an ordinary differential equation (with parameters in A), and the second is an output function for the system.

A *morphism* of (A, B)-continuous dynamical systems $(S_1, f_1^{\mathrm{dyn}}, f_1^{\mathrm{rdt}}) \to (S_2, f_2^{\mathrm{dyn}}, f_2^{\mathrm{rdt}})$ is a smooth map $h \colon S_1 \to S_2$ such that $f_1^{\mathrm{rdt}}(s) = f_2^{\mathrm{rdt}}(hs)$ and $Th(f_1^{\mathrm{dyn}}(a, s)) = f_2^{\mathrm{dyn}}(a, hs)$, where $Th \colon TS_1 \to TS_2$ is the derivative of h.

The category of (A, B)-continuous dynamical systems is denoted by $\mathsf{CDS}(A, B)$. We can now define a functor $\mathsf{CDS} \colon \mathcal{W}_{\mathbf{Euc}} \to \mathbf{Cat}$ as follows. To an object $X = (X^{\mathrm{in}}, X^{\mathrm{out}})$ we assign the category $\mathsf{CDS}(X) := \mathsf{CDS}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$. To a morphism $\phi \colon X \to Y$, we define $\mathsf{CDS}(\phi) \colon \mathsf{CDS}(X) \to \mathsf{CDS}(Y)$ to be the functor which sends a continuous system $(S, f^{\mathrm{dyn}}, f^{\mathrm{rdt}})$ to the continuous system $(S, g^{\mathrm{dyn}}, g^{\mathrm{rdt}})$ where

$$(16) \qquad g^{\mathrm{dyn}}(y, s) := f^{\mathrm{dyn}}\left(\widehat{\phi^{\mathrm{in}}}\big(y, f^{\mathrm{rdt}}(s)\big), s\right) \quad \text{and} \quad g^{\mathrm{rdt}}(s) := \widehat{\phi^{\mathrm{out}}}\big(f^{\mathrm{rdt}}(s)\big)$$

The functor's symmetric lax monoidal structure is again given by cartesian product.

Proposition 2.3.4. *There exists a $\mathcal{W}_{\mathbf{Euc}}$-algebra CDS, for which the labeled boxes inhabitants are continuous dynamical systems.*

Even though discrete and continuous dynamical systems have quite different notions of time and continuity, one can compare (15) and (16) to see that their algebraic structure (e.g. the action on wiring diagrams) is very similar. One aspect of this similarity is summarized as follows.

Proposition 2.3.5. [Spi15, Thm 4.26] *For each $\epsilon > 0$, we have a wiring diagram algebra map*

$$\begin{array}{ccc} \mathcal{W}_{\mathbf{Euc}} & & \\ & \searrow^{\mathsf{CDS}} & \\ \mathcal{W}_U \downarrow \quad \Downarrow \alpha_\epsilon & & \mathbf{Cat} \\ & \nearrow_{\mathsf{DDS}} & \\ \mathcal{W}_{\mathbf{Set}} & & \end{array}$$

where $\mathcal{W}_U$ is as in (10) and α_ϵ is given by Euler's method of linear approximation.

Thus any continuous (A, B)-dynamical system $(S, f^{\mathrm{dyn}}, f^{\mathrm{rdt}})$ gives rise to a discrete (UA, UB) dynamical system $(US, f_\epsilon^{\mathrm{upd}}, f_\epsilon^{\mathrm{rdt}})$ with readout function $f_\epsilon^{\mathrm{rdt}} := U f^{\mathrm{rdt}}$ and with update function given by the linear combination of vectors $f_\epsilon^{\mathrm{upd}}(a, s) := s + \epsilon \cdot (f^{\mathrm{dyn}})_{a,s}$.

2.4. **Spans as $\mathcal{W}$-algebras.** We conclude this section with an explicit construction of a $\mathcal{W}_{\mathcal{C}}$-algebra for any finitely complete category $\mathcal{C}$, which eventually induces the most abstract notion of a machine in Chapter 4. Its special characteristics include the span-like form of the systems inside the labeled boxes, as well as naturally induced algebra maps between such systems.

First of all, recall that if $\mathcal{C}$ is a category with pullbacks, there is a bicategory $\mathbf{Span}_{\mathcal{C}}$ of spans; its objects are the same as $\mathcal{C}$, and hom-categories $\mathbf{Span}_{\mathcal{C}}(X, Y)$ consist of spans $X \nrightarrow Y := X \leftarrow A \rightarrow Y$ in $\mathcal{C}$ as objects, and commutative diagrams

$$
\begin{array}{ccc}
 & A & \\
\swarrow & \downarrow{\scriptstyle\alpha} & \searrow \\
X & & Y \\
\searrow & \downarrow & \nearrow \\
 & B &
\end{array}
$$

as morphisms. Horizontal composition is given by pullbacks, so is associative only up to isomorphism. If moreover $\mathcal{C}$ has finite products, therefore is finitely complete, then spans $X \nrightarrow Y$ can be equivalently viewed as maps $A \rightarrow X \times Y$, and so $\mathbf{Span}_{\mathcal{C}}(X, Y) \cong \mathcal{C}/(X \times Y)$. The following result shows that for such $\mathcal{C}$, we can always define a symmetric lax monoidal functor from $\mathcal{W}_C$ to $\mathbf{Cat}$, using the bicategory of $\mathcal{C}$-spans.

Proposition 2.4.1. *For any finitely complete $\mathcal{C}$, there exists is a $\mathcal{W}_{\mathcal{C}}$-algebra*

$$
\tag{17} \mathsf{Spn}_{\mathcal{C}} \colon \mathcal{W}_{\mathcal{C}} \longrightarrow \mathbf{Cat}
$$

for which labeled boxes inhabitants are spans in $\mathcal{C}$.

Proof. For each box $X = (X^{\mathrm{in}}, X^{\mathrm{out}}) \in \mathrm{ob}\, \mathcal{W}_{\mathcal{C}}$, we define the mapping on objects $\mathsf{Spn}_{\mathcal{C}}(X) := \mathbf{Span}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$ to be the hom-category; it consists of $\mathcal{C}$-spans $S \rightarrow \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$. For each wiring diagram $\phi \colon X \rightarrow Y$ (7), which by (14) determines morphisms in $\mathcal{C}$

$$
\widehat{\phi^{\mathrm{in}}} \colon \widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \rightarrow \widehat{X^{\mathrm{in}}} \qquad \text{and} \qquad \widehat{\phi^{\mathrm{out}}} \colon \widehat{X^{\mathrm{out}}} \rightarrow \widehat{Y^{\mathrm{out}}},
$$

there is a functor $\mathsf{Spn}_{\mathcal{C}}(\phi) \colon \mathbf{Span}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \rightarrow \mathbf{Span}_{\mathcal{C}}(\widehat{Y^{\mathrm{in}}}, \widehat{Y^{\mathrm{out}}})$ mapping some $\widehat{X^{\mathrm{in}}} \leftarrow S \rightarrow \widehat{X^{\mathrm{out}}}$ to the outside span below, formed as the composite (pullback)

(18)

This is clearly functorial, i.e. $\mathsf{Spn}_{\mathcal{C}}(\phi) = (\mathrm{id}, \widehat{\phi^{\mathrm{out}}}) \circ - \circ (\pi_1, \widehat{\phi^{\mathrm{in}}})$ precomposes any span and span morphism with the right and left ϕ-induced spans. Finally, the functor $\mathsf{Spn}_{\mathcal{C}}$ has a symmetric lax monoidal structure by using products symmetry σ in $\mathcal{C}$:

(19)
$$\mathbf{Span}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \times \mathbf{Span}_{\mathcal{C}}(\widehat{Z^{\mathrm{in}}}, \widehat{Z^{\mathrm{out}}}) \xrightarrow{(\mathsf{Spn}_{\mathcal{C}})_{X,Z}} \mathbf{Span}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}} \times \widehat{Z^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}} \times \widehat{Z^{\mathrm{out}}})$$

$$\left(S \xrightarrow{p} \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}, T \xrightarrow{q} \widehat{Z^{\mathrm{in}}} \times \widehat{Z^{\mathrm{out}}} \right) \longmapsto S \times T \xrightarrow{\sigma \circ (p \times q)} \widehat{X^{\mathrm{in}}} \times \widehat{Z^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \times \widehat{Z^{\mathrm{out}}}$$

$\square$

Definition 2.4.2. For $\mathcal{C}$ a finitely complete category, the $\mathcal{W}_C$-algebra $\mathsf{Spn}_{\mathcal{C}}$ described in Proposition 2.4.1 is called the algebra of $\mathcal{C}$-*span systems*.

Explicitly, computing the composite (18) for a $\mathcal{C}$-span $S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$ produces a span $T \to \widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}$ formed by taking the pullback along $\widehat{\phi^{\mathrm{in}}}$ and composing with $\widehat{\phi^{\mathrm{out}}}$:

(20)

$$
\begin{array}{ccc}
T & \longrightarrow & S \\
\downarrow & \lrcorner & \downarrow \\
\widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} & \xrightarrow{(\widehat{\phi^{\mathrm{in}}}, \pi_2)} & \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \\
\downarrow{\scriptstyle 1 \times \widehat{\phi^{\mathrm{out}}}} & & \\
\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}} & &
\end{array}
$$

In particular, if $(f, g) : S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$ is the span on the right and $k : T \to S$ is the top morphism, the left one is (h, gk) inducing the composite $(h, gk\phi^{\mathrm{out}}) : T \to \widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}$.

In fact, since any map $f : A \to B$ in a category $\mathcal{C}$ with pullbacks induces an adjunction $(f_! \dashv f^*)$ between the slice categories, the functor $\mathsf{Spn}_{\mathcal{C}}(\phi) : \mathbf{Span}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \to \mathbf{Span}_{\mathcal{C}}(\widehat{Y^{\mathrm{in}}}, \widehat{Y^{\mathrm{out}}})$ of the above proof can be equivalently expressed as

(21) $$\mathcal{C}/(\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) \xrightarrow{(\widehat{\phi^{\mathrm{in}}}, \pi_2)^*} \mathcal{C}/(\widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) \xrightarrow{(1 \times \widehat{\phi^{\mathrm{out}}})_!} \mathcal{C}/(\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}})$$

between the slice categories.

Remark 2.4.3. It is the case that for any finitely complete category $\mathcal{C}$, the bicategory $\mathbf{Span}_{\mathcal{C}}$ has the structure of a *compact closed bicategory*, see [Sta16]. Since on the 1-category level, every *traced* monoidal category [JSV96] gives rise to a compact closed one, there is evidence that the construction of Proposition 2.4.1 could work if we replaced $\mathbf{Span}_{\mathcal{C}}$ with any traced symmetric monoidal *bicategory* in an appropriate sense.

Following an anonymous reviewer suggestion, for any such bicategory $\mathcal{K}$ where every object is equipped with a comonoid structure, we could define a $\mathcal{W}_{\mathcal{K}}$-algebra as follows:

$$
\begin{array}{ccc}
\mathcal{W}_K & \longrightarrow & \mathbf{Cat} \\
& & \\
(X^{\mathrm{in}}, X^{\mathrm{out}}) & \longmapsto & \mathcal{K}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \\
\phi \downarrow & & \downarrow \Phi \\
(Y^{\mathrm{in}}, Y^{\mathrm{out}}) & \longmapsto & \mathcal{K}(\widehat{Y^{\mathrm{in}}}, \widehat{Y^{\mathrm{out}}})
\end{array}
$$

where Φ maps some $f\colon \widehat{X^{\text{in}}} \to \widehat{X^{\text{out}}}$ to $\Phi(f) = \text{Tr}\left((\widehat{\phi^{\text{out}}} \times 1) \circ \delta \circ f \circ \widehat{\phi^{\text{in}}}\right)$. Explicitly, the map on which the trace acts is

$$\widehat{Y^{\text{in}}} \times \widehat{X^{\text{out}}} \xrightarrow{\widehat{\phi^{\text{in}}}} \widehat{X^{\text{in}}} \xrightarrow{f} \widehat{X^{\text{out}}} \xrightarrow{\delta} \widehat{X^{\text{out}}} \times \widehat{X^{\text{out}}} \xrightarrow{\widehat{\phi^{\text{out}}} \times 1} \widehat{Y^{\text{out}}} \times \widehat{X^{\text{out}}}.$$

Formally defining the structure of a traced symmetric monoidal bicategory, albeit possibly further clarifying the origin of such constructions, is certainly beyond the scope of the current work, and not relevant to the main line. Indeed, central cases of interest are what we will call deterministic and/or total machines, which do not form traced monoidal categories or bicategories. Instead, they will form $\mathcal{W}$-algebras enriched in **Cat**. For relationships between these formalisms, the reader may see [SSR16], which gives a tight relationship between enriched traced monoidal categories algebras on operads similar to $\mathcal{W}$.

Finally, the proposition below shows that the mapping $\mathcal{C} \mapsto \mathsf{Spn}_{\mathcal{C}}$ extends to a functor with target category **WD-Alg**, as described in Section 2.2.

Proposition 2.4.4. *There exists a functor* $\mathsf{Spn}_{(-)}\colon$ **FCCat** $\to$ **WD-Alg** *making the following diagram commute*

(22)

Proof. By Proposition 2.4.1, $\mathsf{Spn}_{\mathcal{C}}\colon \mathcal{W}_{\mathcal{C}} \to$ **Cat** is a symmetric lax monoidal functor. Now given a functor $F\colon \mathcal{C} \to \mathcal{D}$ between typing categories which preserves finite limits, we define a monoidal natural transformation Spn_F as in the diagram below, where $\mathcal{W}_F\colon \mathcal{W}_{\mathcal{C}} \to \mathcal{W}_{\mathcal{D}}$ is a (strong) monoidal functor as in (9):

(23)

Its components $(\mathsf{Spn}_F)_X\colon \mathsf{Spn}_{\mathcal{C}}(X) \to \mathsf{Spn}_{\mathcal{D}}(\mathcal{W}_F(X))$ for each box $X \in \mathcal{W}_{\mathcal{C}}$ are functors that assign to any span $p\colon S \to \widehat{X^{\text{in}}} \times \widehat{X^{\text{out}}}$ the span $(\mathsf{Spn}_F)_X(p) := Fp$, where $Fp\colon FS \to F(\widehat{X^{\text{in}}} \times \widehat{X^{\text{out}}}) \cong F\widehat{X^{\text{in}}} \times F\widehat{X^{\text{out}}}$. Monoidality follows by $F(p \times q) \cong Fp \times Fq$ and (pseudo)naturality follows from the fact that F preserves pullbacks: applying F on the feedback construction (20) is the same as performing that construction on Fp. $\square$

Notice that by construction, if F is a faithful functor, the algebra morphism Spn_F is an embedding, namely has as components functors which are injective on objects and faithful.

This functor $\mathsf{Spn}_{(-)}$ gives rise to wiring diagram algebras of central importance for this work, namely machines (Chapter 4) as well as algebra maps between them (Chapter 5). Moreover, certain subfunctors of it end up capturing fundamental characteristics of systems,

such as totality and determinism (Theorem 5.2.12). What comes next in order to reach these formalisms is the description of the appropriate typing categories, which encompass notions of time.

3. INTERVAL SHEAVES

Guided by some seminal ideas by Lawvere [Law86] and Johnstone [Joh99], in this chapter we describe a site **Int** whose objects can be considered as closed intervals of nonnegative length, and whose morphisms are inclusions of subintervals. A family of subintervals covers an interval if they are jointly surjective on points. We give several examples of sheaves on **Int**, called *interval sheaves*. We then elaborate on variations of this site, giving rise to *discrete* and *synchronous* interval sheaves; these ultimately correspond to different notions of time for our system models at Chapter 4.

In Appendix A we discuss an equivalence between the topos of **Int**-sheaves and the category of *discrete Conduché fibrations* over the additive monoid of nonnegative real numbers, due to Peter Johnstone [Joh99].

3.1. **The interval site Int.** Let $\mathbb{R}_{\geq 0}$ denote the linearly ordered poset of nonnegative real numbers, and for any $p \in \mathbb{R}_{\geq 0}$, let $\mathrm{Tr}_p \colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote the translation-by-p function $\mathrm{Tr}_p(\ell) \coloneqq p + \ell$.

Definition 3.1.1. The category of *continuous intervals*, denoted **Int**, is defined to have:
- objects ob **Int** $= \{\ell \in \mathbb{R}_{\geq 0}\}$,
- morphisms **Int**$(\ell', \ell) = \{\mathrm{Tr}_p \mid p \in \mathbb{R}_{\geq 0} \text{ and } p \leq \ell - \ell'\}$,
- composition $\mathrm{Tr}_p \circ \mathrm{Tr}_{p'} \coloneqq \mathrm{Tr}_{p+p'}$, and
- $\mathrm{id}_\ell = \mathrm{Tr}_0$.

We will sometimes denote Tr_p simply by $p \colon \ell' \to \ell$.

The category **Int** above is the skeleton of (and hence equivalent to) the category whose objects are closed, positively-oriented intervals $[a, b] \in \mathbb{R}$, and whose morphisms are orientation-preserving isometries, i.e. translations by which one interval becomes a subinterval of another. Under the inclusion of the skeleton, an object $\ell \in$ **Int** is sent to the interval $[0, \ell] \subseteq \mathbb{R}$, and we can consider a map $\mathrm{Tr}_p \colon \ell' \to \ell$ as a translation such that $[0, \ell'] \cong [p, p + \ell'] \subseteq [0, \ell]$:

$$\xrightarrow{\hspace{2cm}} $$

$$0 \qquad p \qquad p + \ell' \qquad \ell$$
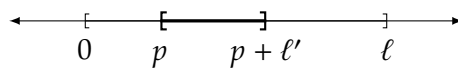
The following analogous definition is obtained by replacing $\mathbb{R}_{\geq 0}$ by the linear order $\mathbb{N}$ of natural numbers.

Definition 3.1.2. The category of *discrete intervals* **Int**$_N$ has as objects the set of natural numbers, ob **Int**$_N = \{n \in \mathbb{N}\}$, as morphisms $n' \to n$ the set of translations Tr_p, where $p \in \mathbb{N}$ and $p \leq n - n'$. The composition and identities are given by $+$ and 0, similarly to Definition 3.1.1.

In what follows, fix $\mathcal{R}$ (resp. $\mathcal{N}$) to be the additive monoid of nonnegative real (resp. natural) numbers, viewed as a category with one object $*$. Proposition 3.1.4 gives a natural

context for the interval categories **Int** and **Int**$_N$ from Definitions 3.1.1 and 3.1.2: they are the twisted arrow categories of $\mathcal{R}$ and $\mathcal{N}$ respectively.

Definition 3.1.3. [Joh99, §2] For any category $\mathcal{C}$, the *twisted arrow category* of $\mathcal{C}$, denoted $\mathcal{C}_{tw}$, has morphisms of $\mathcal{C}$ as objects, and for $f : x \to y$ and $g : w \to z$ in $\mathcal{C}$, a morphism $f \to g$ in $\mathcal{C}_{tw}$ is a pair (u, v) making the diagram commute:

$$
\begin{array}{ccc}
x & \xleftarrow{\;u\;} & w \\
{\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\
y & \xrightarrow[\;v\;]{} & z
\end{array}
$$

Proposition 3.1.4. *There is an isomorphism of categories*

$$\mathbf{Int} \cong \mathcal{R}_{tw} \quad \text{and} \quad \mathbf{Int}_N \cong \mathcal{N}_{tw}.$$

Proof. The objects of $\mathcal{R}_{tw}$ are nonnegative real numbers $\ell : * \to *$, and a morphism $\ell' \to \ell$ in $\mathcal{R}_{tw}$ is a pair of nonnegative real numbers (p, q) such that $p + \ell' + q = \ell$. But q is completely determined by p, ℓ, and ℓ', hence we can identify (p, q) with the morphism $\mathrm{Tr}_p : \ell' \to \ell$ of Definition 3.1.1. Similarly for the discrete intervals. $\qquad\square$

Finally, the one-object categories $\mathcal{R}$ and $\mathcal{N}$ themselves fall under the general description of a factorization-linear category defined below. In particular, $\mathrm{Fact}(q)$ is called the *interval category* in [Law86] and denoted by $\llbracket q \rrbracket$ in [Joh99].

Definition 3.1.5. For any category $\mathcal{C}$ and morphism $q : a \to b$ in $\mathcal{C}$, define the *q-factorization category*, denoted $\mathrm{Fact}(q)$, as follows. Its objects are triples $M = (m, f, g)$, where $f : a \to m$ and $g : m \to b$ are arrows in $\mathcal{C}$ composing to $q = g \circ f$. If $M' = (m', f', g')$, a morphism $M \to M'$ in $\mathrm{Fact}(q)$ is an arrow $k : m \to m'$ in $\mathcal{C}$ making the following triangles commute

$$
\begin{array}{ccc}
a & \xrightarrow{\;f'\;} & m' \\
{\scriptstyle f}\downarrow & {\scriptstyle k}\nearrow \quad {\scriptstyle q} & \downarrow{\scriptstyle g'} \\
m & \xrightarrow[\;g\;]{} & b
\end{array}
$$

We furthermore say that $\mathcal{C}$ is *factorization-linear* if for every morphism q, $\mathrm{Fact}(q)$ is a linear preorder, i.e. if for any pair of objects $M, M' \in \mathrm{Fact}(q)$,
  (1) there is at most one morphism $M \to M'$, and
  (2) there exists either a morphism $M \to M'$ or a morphism $M' \to M$.

Example 3.1.6.
  (1) For any directed graph G, the free category $\mathcal{C} = \mathsf{Fr}(G)$ on G is factorization-linear. Since $\mathcal{N}$ is the free category on the terminal graph, it is factorization-linear.
  (2) The category $\mathcal{R}$ is also factorization-linear. Indeed, if f, g, f', g' are nonnegative real numbers with $f + g = f' + g'$, then there is exactly one real number k such that $k + f = f'$ (iff $g' + k = g$), and $k \geq 0$ iff there is a map $(f, g) \to (f', g')$.

We will later employ these characterizations in order to consider sheaves on the categories of intervals. Below we fix some notation for presheaves on them.

Notation 3.1.7. For any small category $\mathcal{C}$ (such as **Int** or **Int**$_N$), we denote by $\mathsf{Psh}(\mathcal{C}) = [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ the category of presheaves. Consider an **Int**-presheaf $A\colon \mathbf{Int}^{\mathrm{op}} \to \mathbf{Set}$. For any continuous interval ℓ, we refer to elements $x \in A(\ell)$ as *sections of A on ℓ*. We refer to sections of length $\ell = 0$ as *germs of A*. For any section $x \in A(\ell)$ and any map $\mathrm{Tr}_p\colon \ell' \to \ell$, we write $x|_{[p,p+\ell']}$ to denote the restriction $A(\mathrm{Tr}_p)(x) \in A(\ell')$. Similar notation applies to **Int**$_N$.

We can graphically depict a section x on a continuous interval $\ell \in \mathbf{Int}$, together with a restriction, and a section y on a discrete interval $n \in \mathbf{Int}_N$, as
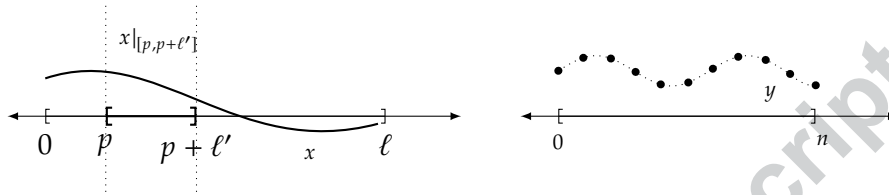
Certain classes of restrictions will come up often, so we create some special notation for them. For any $0 \le \ell' \le \ell$, we write $\lambda_{\ell'}\colon A(\ell) \to A(\ell')$ and $\rho_{\ell'}\colon A(\ell) \to A(\ell')$ to denote the restrictions along the maps $[0, \ell'] \subseteq [0, \ell]$ and $[\ell - \ell', \ell] \subseteq [0, \ell]$ respectively. We refer to them as the *left* and *right* restrictions of length ℓ'. In particular if $\ell' = 0$, we will often use the notation

$$\lambda_0(x) := x|_{[0,0]} \in A(0) \qquad \text{and} \qquad \rho_0(x) := x|_{[\ell,\ell]} \in A(0)$$

to denote the left and right endpoint germs of x.

3.2. **Int-sheaves.** We now wish to equip each of the categories **Int** and **Int**$_N$ with a coverage. In [Joh99], Peter Johnstone defined such a coverage for all twisted arrow categories $\mathcal{C}_{\mathsf{tw}}$ with $\mathcal{C}$ factorization-linear. For such $\mathcal{C}$, he moreover proved that the category of sheaves on $\mathcal{C}_{\mathsf{tw}}$ is equivalent to the category of discrete Conduché fibrations over $\mathcal{C}$, a brief discussion on which we postpone until Appendix A; Johnstone's result is Theorem A.2.1.

Definition 3.2.1. Let $\mathcal{C}$ be a factorization-linear category, Definition 3.1.5, and $\mathcal{C}_{\mathsf{tw}}$ its twisted arrow category, Definition 3.1.3. The *Johnstone coverage* on $\mathcal{C}_{\mathsf{tw}}$ is defined as follows: a covering family for an object $(h\colon x \to z) \in \mathcal{C}_{\mathsf{tw}}$ is any pair of maps $(\mathrm{id}_x, g)\colon f \to h$ and $(f, \mathrm{id}_z)\colon g \to h$ in $\mathcal{C}_{\mathsf{tw}}$, where $g \circ f = h$ in $\mathcal{C}$,

It can be checked that this is indeed a coverage [Joh99, §3.6], and we refer to the associated site as the *Johnstone site* for $\mathcal{C}_{\mathsf{tw}}$.[6]

Applying the above definition to $\mathcal{R}$ and $\mathcal{N}$, which are factorization-linear by Example 3.1.6, we obtain the Johnstone sites **Int** and **Int**$_N$ on their twisted-arrow categories, see Proposition 3.1.4. Below we explicitly describe them, and fix our terminology.

---

[6]In fact, [Joh99, 3.6] explains that $\mathcal{C}$ being factorization linear is stronger than necessary to define a coverage on $\mathcal{C}_{\mathsf{tw}}$. However, this stronger condition is necessary for Corollary A.2.2 to hold.

Definition 3.2.2. For any interval $\ell \in \mathbf{Int}$, and any real number $p \in [0, \ell]$, we say that the pair of subintervals $[0, p]$ and $[p, \ell]$, or more precisely the morphisms $\mathrm{Tr}_0 \colon p \to \ell$ and $\mathrm{Tr}_p \colon (\ell - p) \to \ell$, form a cover, which we call the *p-covering family* for ℓ.
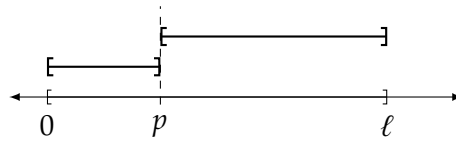
Similarly, the pairs $([0, p], [p, n])$ for any natural number $p \in \{0, \ldots, n\}$ form the *p-covering family* for $n \in \mathbf{Int}_N$.

If X is an $\mathbf{Int}$-presheaf, we say that sections $x_1 \in X(\ell_1)$ and $x_2 \in X(\ell_2)$ are *compatible* if the right endpoint of x_1 matches the left endpoint of x_2, i.e. if $\rho_0(x_1) = \lambda_0(x_2)$. Thus X satisfies the sheaf axiom for the above coverage if, whenever x_1 and x_2 are compatible, there is a unique section $x_1 * x_2 \in X(\ell_1 + \ell_2)$, called the *gluing* of x_1 and x_2, such that $\lambda_{\ell_1}(x_1 * x_2) = x_1$ and $\rho_{\ell_2}(x_1 * x_2) = x_2$.

We refer to an $\mathbf{Int}$-presheaf satisfying the sheaf axiom as a *continuous interval sheaf*, or simply $\mathbf{Int}$-*sheaf*. We denote by $\widetilde{\mathbf{Int}}$ the full subcategory of sheaves; it has the structure of a topos. Similarly, we define *discrete interval sheaves* or $\mathbf{Int}_N$-*sheaves* and the topos $\widetilde{\mathbf{Int}}_N$ thereof.

Each of the inclusions $U \colon \widetilde{\mathbf{Int}} \hookrightarrow \mathsf{Psh}(\mathbf{Int})$ and $\widetilde{\mathbf{Int}}_N \hookrightarrow \mathsf{Psh}(\mathbf{Int}_N)$ has a left adjoint. In each case we denote it asSh and call it the *associated sheaf* or *sheafification* functor.

The category of $\mathbf{Int}$-sheaves is certainly more complex than that of $\mathbf{Int}_N$-sheaves; indeed the latter is just the category of graphs.

Proposition 3.2.3. *There is an equivalence of categories $\widetilde{\mathbf{Int}}_N \simeq \mathbf{Grph}$, between the category of discrete interval sheaves and the category of graphs.*

Proof. To every sheaf $X \in \widetilde{\mathbf{Int}}_N$, we associate the graph $G = (V, E, \mathrm{src}, \mathrm{tgt})$ with $V := X(0)$, $E := X(1)$, $\mathrm{src} := X(\mathrm{Tr}_0)$, and $\mathrm{tgt} := X(\mathrm{Tr}_1)$, where $\mathrm{Tr}_0, \mathrm{Tr}_1 \colon 0 \to 1$ are the two inclusions; see Definition 3.1.2. To every graph G we associate the $\mathbf{Int}_N$-sheaf $\mathrm{Path}(G)$ for which $\mathrm{Path}(G)(n)$ is the set of length-n paths in G with the obvious restriction maps. This is indeed a sheaf because two paths match if the ending vertex of one is the starting vertex of the other, in which case the paths can be concatenated.

The above constructions are clearly functorial; we need to check that these functors are mutually inverse. The roundtrip functor for a graph G returns a graph with the length-0 and length-1 paths in G, which is clearly isomorphic to G. For an $\mathbf{Int}_N$-sheaf X, the roundtrip is again X because the coverage on $\mathbf{Int}_N$ ensures that every section is completely determined by the length-0 and length-1 data.   □

Remark 3.2.4. Proposition 3.2.3 is closely related to the classical nerve theorem for categories, as presented by [Ber02] and generalized by [Lei04b, Web07, BMW12]; see [Koc11, §2.01] for a clear and concise summary of this approach to nerve theorems.

Our category $\mathbf{Int}_N$ is equivalent to the subcategory—written Δ_0 in [Koc11]—of the simplicial category Δ containing all objects and all 'free' morphisms (called *immersions* in [Ber02]). A morphism $\phi \colon [m] \to [n]$ is free if $\phi(i + 1) = \phi(i) + 1$, i.e. if ϕ is distance

preserving. In those sources, it is shown that the classical Segal condition characterizing which simplicial sets arise as the nerve of some category can be nicely phrased in terms of Δ_0: a presheaf $f\colon \Delta \to \mathbf{Set}$ is the nerve of a category if and only if the restriction of f to Δ_0 is a sheaf for a certain coverage, and moreover the category of sheaves for this coverage on Δ_0 is equivalent to $\mathbf{Grph}$. This site Δ_0 is equivalent to the Johnstone site $\mathbf{Int}_N$. We thank one of our reviewers for pointing out this connection.

Example 3.2.5. We do not include numerous examples of $\mathbf{Int}_N$-sheaves, assuming the reader is familiar with graphs; the following construction will be used later.

For any set S, consider the complete graph $K(S) := (S \times S \rightrightarrows S) \in \mathbf{Grph}$, having a vertex for each $s \in S$ and an edge from $s \to s'$ for each $(s, s') \in S \times S$. By Proposition 3.2.3, there is a canonical way to view any graph G as an $\mathbf{Int}_N$-sheaf, under which $K(S)$ becomes the $\mathbf{Int}_N$-sheaf whose length-n sections are S-lists $\langle s_0, \ldots, s_n \rangle$ of length $n + 1$, namely $K(S)(n) = S^{n+1}$. This induces a functor $K\colon \mathbf{Set} \to \widetilde{\mathbf{Int}_N}$.

Example 3.2.6. We collect some useful examples of $\mathbf{Int}$-sheaves.

(1) For any $\ell \in \mathbf{Int}$, we have the representable presheaf $\mathsf{Yon}_\ell = \mathrm{Hom}_{\mathbf{Int}}(-, \ell)$. For each object ℓ' we have $\mathsf{Yon}_\ell(\ell') \cong \{p \in \mathbb{R}_{\geq 0} \mid p \leq \ell - \ell'\}$, so in particular $\mathsf{Yon}_\ell(\ell') = \emptyset$ if $\ell' \geq \ell$. For each morphism $\mathrm{Tr}_q\colon \ell' \to \ell$, we have $\mathsf{Yon}_\ell(p) = p + q$. The coverage on $\mathbf{Int}$ is sub-canonical, meaning that for any ℓ, the representable presheaf is in fact a sheaf, i.e. $\mathsf{Yon}_\ell \in \widetilde{\mathbf{Int}}$. One can think of Yon_ℓ as the sheaf whose length ℓ' sections are all 'placements of a length ℓ' interval inside of a length ℓ interval'.

(2) For any set S, the sheaf of functions $\mathsf{Fnc}(S)$ assigns to each interval ℓ the set of functions $\mathsf{Fnc}(S)(\ell) = \{f\colon [0, \ell] \to S\}$, with the obvious restriction maps. The functor $\mathsf{Fnc}\colon \mathbf{Set} \to \widetilde{\mathbf{Int}}$ is a right adjoint, see Proposition 3.2.7.

(3) For any set S, the constant sheaf $\mathrm{cnst}(S)$ is defined by $\mathrm{cnst}(S)(\ell) := S$ for any $\ell \in \mathbf{Int}$, and identity restrictions maps. The functor $\mathrm{cnst}\colon \mathbf{Set} \to \widetilde{\mathbf{Int}}$ has both a right adjoint $\widetilde{\mathbf{Int}}(1, -)\colon \widetilde{\mathbf{Int}} \to \mathbf{Set}$ and a left adjoint $\pi_0\colon \widetilde{\mathbf{Int}} \to \mathbf{Set}$, given by $\pi_0(X) := X(0)/\!\!\sim$, where $\sim$ is the equivalence relation generated by $\lambda_0(x) \sim \rho_0(x)$ for any ℓ and $x \in X(\ell)$.

(4) If X is a C^n-manifold, there is a sheaf of C^n curves in X; we just need to be careful with the length-0 sections. Define a sheaf $C^n(X)$ of C^n curves through X by

(24)
$$C^n(X)(\ell) := \left\{ (\epsilon, f) \mid \epsilon > 0, f \in C^n\big((-\epsilon, \ell + \epsilon), X\big) \right\} / \!\!\sim$$

where we set $(\epsilon, f) \sim (\epsilon', f')$ if one is the restriction of the other, say $f'|_{(-\epsilon, \ell+\epsilon)} = f$.

For $\ell > 0$, the closed interval $[0, \ell]$ is a manifold with boundary, so in fact the isomorphism $C^n(X)(\ell) \cong \{f\colon [0, \ell] \to X \mid f \in C^n\}$ is the usual definition of continuously n-times differentiable functions out of $[0, \ell]$. However, (24) makes sense when $\ell = 0$ also: it defines the set of n-jets in X. It should be emphasized that the length-0 sections of $C^n(X)$ are not the points in X, so the notation $C^n(X)$ could be considered misleading. However, it is the right definition to define a sheaf of C^n curves, because gluing two C^n curves whose endpoints have the same n-jet results in a C^n curve.

Also notice that if $m \leq n$, there is a sheaf morphism $C^n(X) \to C^m(X)$.

(5) Let $0, 1\colon \mathsf{Yon}_0 \to \mathsf{Yon}_1$ denote the image under $\mathsf{Yon}\colon \mathbf{Int} \to \widetilde{\mathbf{Int}}$ of the left and right endpoint inclusions $\mathsf{Tr}_0, \mathsf{Tr}_1\colon 0 \to 1$ in $\mathbf{Int}$. Then we define the *periodic synchronizing sheaf* to be the quotient $\mathsf{Sync} \coloneqq \mathsf{Yon}_1/(0 = 1)$. For an interval ℓ, we may identify

(25) $$\mathsf{Sync}(\ell) \cong \mathbb{R}/\mathbb{Z} \cong \{0 \leq [\theta] < 1\}$$

where $[\theta]$ denotes the equivalence class modulo $\mathbb{Z}$ of $\theta \in \mathbb{R}$ and may be called the *phase*. One can imagine a length-ℓ section of Sync as a helix of height ℓ; two different sections differ only by their phase (turning the helix by some angle θ). Given a subinterval $\mathsf{Tr}_p\colon \ell' \to \ell$, we have $\mathsf{Sync}(\mathsf{Tr}_p)(\theta) = [\theta + p]$. That is, we simply restrict the helix to the subinterval $[p, p + \ell'] \subseteq [0, \ell]$. See Remark 3.3.1 for how this sheaf arises in our framework.

Some of the above constructions are naturally connected in the following way.

Proposition 3.2.7. *Consider the functor* $\mathsf{Stk}_0\colon \widetilde{\mathbf{Int}} \to \mathbf{Set}$, *given by* $\mathsf{Stk}_0(A) \coloneqq A(0)$. *It is the inverse image of an essential geometric morphism*

$$\mathbf{Set} \xrightarrow[\xleftarrow{\quad \mathsf{Stk}_0 \quad}]{\overset{(-)\cdot\mathsf{Yon}_0}{\longrightarrow}} \widetilde{\mathbf{Int}}$$
$$\underset{\mathsf{Fnc}}{\longrightarrow}$$

i.e. Stk_0 *is an essential point of the topos* $\widetilde{\mathbf{Int}}$.

Proof. The right adjoint of Stk_0 is $\mathsf{Fnc}(-)$ defined in Example 3.2.6(2). The left adjoint of Stk_0 sends S to the copower $S \cdot \mathsf{Yon}_0 \in \widetilde{\mathbf{Int}}$, where Yon_0 is as in Example 3.2.6(1). □

The following notions of *extension* for a section and a sheaf will be of central importance in Chapter 4.

Definition 3.2.8. Let $A \in \widetilde{\mathbf{Int}}$ be a sheaf, and $\epsilon \geq 0$. For a section $a \in A(\ell)$, if $a' \in A(\ell + \epsilon)$ is a section with $a'|_{[0,\ell]} = a$, we call a' an *ϵ-extension of a*. Moreover, we define the *ϵ-extension sheaf* $\mathsf{Ext}_\epsilon(A)$ of A by assigning to ℓ the set

$$\mathsf{Ext}_\epsilon(A)(\ell) \coloneqq A(\ell + \epsilon)$$

induced by the functor $\mathbf{Int} \to \mathbf{Int}$ sending $\ell \mapsto \ell + \epsilon$ and $\mathsf{Tr}_p \mapsto \mathsf{Tr}_p$.

Essentially, the extension sheaf of A includes, for each interval of time, information (i.e. sections) for ϵ-longer intervals; this will be essential when discussing feedback of systems, later formalized e.g. by Definition 4.2.4.

The above defines an endofunctor $\mathsf{Ext}_\epsilon\colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}$[7]. There exist two natural transformations $\lambda, \rho\colon \mathsf{Ext}_\epsilon \Rightarrow \mathrm{id}_{\widetilde{\mathbf{Int}}}$ whose components $\mathsf{Ext}_\epsilon(A) \to A$ are given by applying either left or right restriction as in Notation 3.1.7,

(26) $$\mathsf{Ext}_\epsilon A(\ell) = A(\ell + \epsilon) \xrightarrow[\rho_\ell]{\lambda_\ell} A(\ell).$$

That is, for $x \in A(\ell + \epsilon)$ we have $\lambda(x) \coloneqq \lambda_\ell(x) = x|_{[0,\ell]}$ and $\rho(x) \coloneqq \rho_\ell(x) = x|_{[\epsilon, \ell+\epsilon]}$. Similarly for a discrete-interval sheaf $A \in \widetilde{\mathbf{Int}}_N$ and $n \in \mathbb{N}$, we define the extension sheaf $\mathsf{Ext}_n(A)$ and the natural transformations λ, ρ with components $\mathsf{Ext}_n(A) \to A$.

We finish this section by collecting a few useful lemmas.

---

[7]It should be clear that this functor Ext has nothing to do with the Ext functor from homological algebra.

Lemma 3.2.9. *For any $\epsilon \geq 0$ the functor* $\mathsf{Ext}_\epsilon \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}$ *commutes with all limits.*

Proof. Limits in $\widetilde{\mathbf{Int}}$ are taken pointwise. □

Lemma 3.2.10. *Suppose given a commutative triangle of* $\widetilde{\mathbf{Int}}$*-sheaves*

$$
\begin{array}{ccc}
\mathsf{Ext}_\epsilon S & \xrightarrow{\ h\ } & S' \\
& \searrow{\scriptstyle \lambda} & \downarrow{\scriptstyle k} \\
& & S
\end{array}
$$

Then h is an epimorphism (monomorphism) if and only if all components of the underlying presheaf morphism Uh are surjective (injective).

Proof. First of all, the functor $U \colon \widetilde{\mathbf{Int}} \subseteq \mathsf{Psh}(\mathbf{Int})$ is faithful, so if h is any sheaf map and Uh is an epimorphism (monomorphism) then so is h. Moroever, since U is right adjoint to the sheafification asSh, it preserves monomorphisms.

Now suppose h is an epimorphism; we will show that h_ℓ is a surjection, for any $\ell \in \mathbf{Int}$, so choose $s' \in S'(\ell)$. Recall that h being an epimorphism means that there exists a cover $0 = \ell_0 \leq \cdots \leq \ell_n = \ell$ such that each restriction $s_i' \coloneqq s'|_{[\ell_i, \ell_{i+1}]} = h_{\ell_{i+1} - \ell_i}(s_i)$ for some $s_i \in \mathsf{Ext}_\epsilon S(\ell_{i+1} - \ell_i)$. We may assume $n = 2$. Since the s_i' are compatible in S' and since $\lambda(s_i) = k(s_i')$, we have that $\lambda(s_1)$ and $\lambda(s_2)$ are compatible sections of S. Thus we can glue $s \coloneqq \lambda(s_1) * s_2 \in S(\ell_2 + \epsilon)$, giving $s \in \mathsf{Ext}_\epsilon S(\ell)$ with $h_\ell(s) = s'$. □

3.3. **Synchronization.** For what follows, it is essential that we compare the toposes $\widetilde{\mathbf{Int}}_N$ and $\widetilde{\mathbf{Int}}$ as the far ends of the time spectrum: from sections over continuous intervals of time to those over specific, equally-spaced ticks of the clock. The following remark discusses why the notion of a *synchronous* sheaf is required for a coherent common time-framework.

Remark 3.3.1. Suppose X is a discrete-interval sheaf and we want to translate it into a continuous-interval sheaf X'. What should the X'-sections of length 1.5 be? Taking them to be $X(1)$ does not give rise to any reasonable sort of restriction map. Instead, the adjoint to the obvious forgetful functor from continuous to discrete sheaves, denoted Σ_i in Proposition 3.3.3, naturally adds a notion of *phase*—a number between 0 and 1—so as to enable arbitrary real-number restrictions. In particular, $\Sigma_i(\mathbf{1})$ of the terminal discrete sheaf is the sychronizing sheaf of Example 3.2.6,(5). Thus while there is a left adjoint functor from discrete- to continuous-time sheaves that preserves pullbacks (because sections of matching phases can be glued), it does not preserve the terminal object: this means it can be replaced by a geometric morphism into the slice topos $\widetilde{\mathbf{Int}}/\Sigma_i(\mathbf{1})$, which can later be used as a type-changing functor (9) for wiring diagram algebras.

Let us explicitly establish the above observations. If $i \colon \mathbf{Int}_N \to \mathbf{Int}$ is the evident inclusion, it induces an adjunction between the respective toposes of sheaves as follows.

Proposition 3.3.2. *The functor $i \colon \mathbf{Int}_N \to \mathbf{Int}$ preserves covering families.*

Proof. This follows directly from Definition 3.2.2. □

Let $i^* \colon \mathsf{Psh}(\mathbf{Int}) \to \mathsf{Psh}(\mathbf{Int}_N)$ denote the functor given by pre-composing a presheaf $X \colon \mathbf{Int}^{\mathrm{op}} \to \mathbf{Set}$ with i^{op}, and let asSh denote the sheafification as in Definition 3.2.2; the following fact is well-known.

Proposition 3.3.3. *The functor $i \colon \mathbf{Int}_N \to \mathbf{Int}$ induces an adjunction $\Sigma_i \dashv \Delta_i$ between sheaf toposes*

(27)

$$
\begin{array}{ccc}
\widetilde{\mathbf{Int}}_N & \xrightarrow{\;\;\Sigma_i\;\;} & \widetilde{\mathbf{Int}} \\[2pt]
 & \xleftarrow[\;\;\Delta_i\;\;]{\perp} & \\
\mathsf{asSh} \dashv U \uparrow\downarrow & & \mathsf{asSh} \dashv U \uparrow\downarrow \\
\mathsf{Psh}(\mathbf{Int}_N) & \xrightarrow{\;\;\mathrm{Lan}_i\;\;} & \mathsf{Psh}(\mathbf{Int}) \\[2pt]
 & \xleftarrow[\;\;i^*\;\;]{\perp} &
\end{array}
$$

such that $U \circ \Delta_i = i^* \circ U$, hence also $\Sigma_i \circ \mathsf{asSh} \cong \mathsf{asSh} \circ \mathrm{Lan}_i$.

Proof. Because i preserves covers (Proposition 3.3.2), the composite $\widetilde{\mathbf{Int}} \xrightarrow{U} \mathsf{Psh}(\mathbf{Int}) \xrightarrow{i^*} \mathsf{Psh}(\mathbf{Int}_N)$ factors through $\widetilde{\mathbf{Int}}_N$, defining Δ_i. Since U is fully faithful and has a left adjoint asSh, it can be seen that the composite $\Sigma_i \coloneqq \mathsf{asSh} \circ \mathrm{Lan}_i \circ U$ is left adjoint to Δ_i. $\qquad\square$

We will see in Corollary 3.3.6 that $\Sigma_i \dashv \Delta_i$ is not in general a geometric morphism because the left adjoint Σ_i does not preserve the terminal object (nor binary products). However, what is important for our purposes is to see that it preserves pullbacks, thus is a pre-geometric morphism in the sense of [Joh02, A4.1.13]. This fact will be deduced from a direct formula calculation for Σ_i as a coproduct (denoted using $\sqcup$). For any $x \in \mathbb{R}$, let $\lfloor x \rfloor, \lceil x \rceil \in \mathbb{Z}$ denote the floor (resp. ceiling) of x, i.e. the largest integer $\lfloor x \rfloor \le x$ (resp. the smallest integer $\lceil x \rceil \ge x$).

Proposition 3.3.4. *For any discrete-interval sheaf $X \in \widetilde{\mathbf{Int}}_N$, there is an isomorphism*

(28)
$$
\Sigma_i X(\ell) \cong \bigsqcup_{r \in [0,1)} X\big(\lceil r + \ell \rceil\big)
$$

Moreover, Σ_i commutes with the forgetful functor U in (27), i.e. $U \circ \Sigma_i \cong \mathrm{Lan}_i \circ U$.

Proof. The pointwise left Kan extension $C \coloneqq \Sigma_i X = \mathrm{Lan}_i X$ is computed as a colimit

(29)
$$
C(\ell) = \operatorname*{colim}_{\substack{n \in \mathbf{Int}_N \\ (i(n) \to \ell) \in \mathbf{Int}^{\mathrm{op}}}} X(n) = \int^{n \in \mathbf{Int}_N} \mathbf{Int}^{\mathrm{op}}\big(i(n), \ell\big) \times X(n).
$$

The indexing category $(i^{\mathrm{op}} \downarrow \ell)$ is a poset with the property that every element maps to a unique maximal element. Such maximal elements correspond to maps $r \colon \ell \to n$ in $\mathbf{Int}$, for which $0 \le r < 1$ and $0 \le n - (r + \ell) < 1$, i.e. $n = \lceil r + \ell \rceil$. Thus for each ℓ, the set $C(\ell)$ is exactly as in the formula (28): an element of $C(\ell)$ is a pair (r, x), where $0 \le r < 1$ and $x \in X(n)$. We need to show that C is a sheaf, but to do so we must better understand the restriction maps; see Example 3.3.5 for intuition.

So suppose given (r, x) as above. Given a map $p \colon \ell' \to \ell$, let $a \coloneqq \lfloor p + r \rfloor$, $b \coloneqq \lceil p + r + \ell' \rceil$, $n' \coloneqq b - a$, and $r' \coloneqq p + r - a$, so $0 \le r' < 1$. We have $a \colon n' \to n$ in $\mathbf{Int}_N$ such that $r \circ p = a \circ r'$. Given a section $x \in X\big(\lceil r + \ell \rceil\big)$, the restriction map for C is given by

(30)
$$
C(p)(r, x) = \big(r', x|_{[a,b]}\big).
$$

Suppose given sections $(r, x) \in C(\ell)$ and $(r', x') \in C(\ell')$, where $x \in X(n)$ and $x' \in X(n')$. If they are compatible then, by the above restriction formula (30), we must have $r' =$

$\ell + r - \lfloor \ell + r \rfloor$ and

$$x\big|_{\left[\lfloor \ell+r \rfloor, \lceil \ell+r \rceil\right]} = (r, x)|_{[\ell, \ell]} = (r', x')|_{[0,0]} = x'\big|_{\left[\lfloor r' \rfloor, \lceil r' \rceil\right]}.$$

If we denote by n_0 the length of this section, then n_0 is either 0 or 1, depending on whether $\ell + r = \lfloor \ell + r \rfloor$ or not. Either way, x and x' are compatible sections of X and can be glued to form $x * x' \in X(n - n_0 + n')$. Thus we have shown that $C = \mathrm{Lan}_i X$ is a sheaf. It follows that $U \circ \mathsf{asSh} \circ \mathrm{Lan}_i = \mathrm{Lan}_i$, so $U \circ \Sigma_i = \mathrm{Lan}_i \circ U$. $\qquad\square$

Example 3.3.5. Given a discrete sheaf X, a length ℓ section of the continuous sheaf $C := \Sigma_i(X)$ is given by choosing a phase $r \in [0, 1)$ and an element $x \in X(n)$, where $n := \lceil r + \ell \rceil$. Thus x is a section of X on the smallest discrete interval containing the continuous interval, when embedded to start at r. In black is a picture of a map $r \colon \ell \to n$ in **Int**, where $n = 6$, $r = \frac{2}{3}$, $\ell = \frac{29}{6}$, and $r + \ell = \frac{11}{2}$:
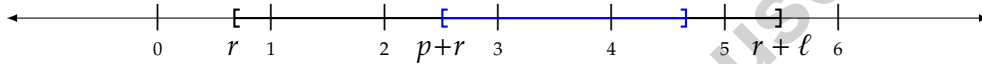
therefore a section in $C\left(\frac{29}{6}\right)$ could be a pair $\left(\frac{2}{3}, x \in X(6)\right)$.

Suppose we want to restrict $\left(\frac{2}{3}, x\right)$ to the blue interval. In this case we visually we see that $p + r = 2.5$, so we first take the new phase to be its fractional part, $r' := 0.5$. We then restrict x to the smallest discrete interval containing the blue interval when embedded to start at 0.5, i.e. $x|_{[2,5]}$.

The periodic synchronizing sheaf Sync from Example 3.2.6(5) arises as Σ_i of the terminal $\mathbf{Int}_N$-sheaf.

Corollary 3.3.6. *The functor Σ_i preserves pullbacks but not the terminal object; indeed we have*

$$\Sigma_i(\mathrm{cnst}\{*\}) \cong \mathsf{Sync}.$$

Proof. The fact that Σ_i preserves pullbacks can be checked by hand from (28); formally, it is due to the fact that the indexing category $(i^{\mathrm{op}} \downarrow \ell)$ in (29) is a coproduct of filtered categories, see [ABLR02, Ex. 1.3(vi)].

The fact that $\Sigma_i(\mathrm{cnst}\{*\}) \cong \mathsf{Sync}$ can again be checked by hand, e.g. its set of length ℓ sections is given by the coproduct $\bigsqcup_{r \in [0,1)} 1 \cong [0, 1)$. Another approach is to note that Σ_i can be regarded as a functor $\Sigma_i \colon \mathbf{Grph} \to \widetilde{\mathbf{Int}}$ by Proposition 3.2.3. Using (29) one can check that the image of the single-vertex graph $\boxed{\bullet}$ is the sheaf Yon_0 and that the image of the walking edge graph $E = \boxed{\bullet^0 \to \bullet^1}$ is Yon_1, from Example 3.2.6(1). The terminal $\mathbf{Int}_N$-sheaf $\mathbf{1} = \mathrm{cnst}\{*\}$ corresponds to the quotient graph $E/(0 = 1)$. Since Σ_i preserves colimits, we indeed have $\Sigma_i(\mathrm{cnst}\{*\}) \cong \mathsf{Sync}$. $\qquad\square$

Our next goal is to compare the discrete-interval sheaf topos $\widetilde{\mathbf{Int}}_N$ with the slice topos $\widetilde{\mathbf{Int}}/\mathsf{Sync}$ over the synchronizing sheaf (25), whose objects $X \to \mathsf{Sync}$ we sometimes call *synchronous sheaves*. An object $X \to \mathsf{Sync}$ of this should be visualized as an ordinary **Int**-sheaf X, where every section of X has been assigned a section of Sync, i.e. a helix at some phase $\theta \in [0, 1)$ or a watch-hand at some position. Two sections x, y are 'in sync' if they both have the same phase.

Proposition 3.3.7. *The adjunction $\Sigma_i \dashv \Delta_i$ between $\widetilde{\mathbf{Int}}$ and $\widetilde{\mathbf{Int}}_N$ from Proposition* 3.3.3 *factors through a geometric morphism*

$$\tag{31} \widetilde{\mathbf{Int}}_N \; \underset{\Delta_i'}{\overset{\Sigma_i'}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \; \widetilde{\mathbf{Int}}/\mathrm{Sync}$$

in which the left adjoint Σ_i' is fully faithful.

Proof. Define a functor Σ_i' which sends $X \in \widetilde{\mathbf{Int}}_N$ to the synchronous sheaf $\Sigma_i(X \overset{!}{\to} \mathbf{1})$. Because Σ_i preserves pullbacks, it can be verified that Σ_i' preserves all finite limits.

We now construct its right adjoint. First note that $\Delta_i \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}_N$ is given by pre-composing with the inclusion $i \colon \mathbf{Int}_N \to \mathbf{Int}$. Applying it to Sync results in the constant sheaf $\Delta_i(\mathrm{Sync}) \cong \mathrm{cnst}(\mathbb{R}/\mathbb{Z}) \in \widetilde{\mathbf{Int}}_N$; indeed, it is constant because restricting Sync along maps in $\mathbf{Int}_N$ does not change the phase. Hence, given a synchronous sheaf $Y \to \mathrm{Sync}$, define $\Delta_i'(Y)$ to be the following pullback in $\mathbf{Int}_N$:

$$
\begin{array}{ccc}
\Delta_i'(Y) & \longrightarrow & \Delta_i(Y) \\
\downarrow & \lrcorner & \downarrow \\
\mathrm{cnst}\{*\} & \longrightarrow & \Delta_i(\mathrm{Sync})
\end{array}
$$

where the bottom map is the unit $\mathrm{cnst}\{*\} \to \Delta_i \Sigma_i(\mathrm{cnst}\{*\})$; see Corollary 3.3.6. Since $\mathrm{cnst}\{*\} \in \widetilde{\mathbf{Int}}_N$ is a terminal object, it follows immediately that Δ_i' is right adjoint to Σ_i'.

The adjunction $\Sigma_i \dashv \Delta_i$ can be verified to be the composite of $\Sigma_i' \dashv \Delta_i'$ with the adjunction

$$\tag{32} \widetilde{\mathbf{Int}}/\mathrm{Sync} \; \underset{\Delta_{\mathrm{Sync!}}}{\overset{\Sigma_{\mathrm{Sync!}}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \; \widetilde{\mathbf{Int}}$$

between slice categories as induced by the unique map $\mathrm{Sync!} \colon \mathrm{Sync} \to \{*\}$ in $\widetilde{\mathbf{Int}}$. That is, $\Sigma_{\mathrm{Sync!}}(X \to \mathrm{Sync}) = X$ and $\Delta_{\mathrm{Sync!}}(Y) = (Y \times \mathrm{Sync} \to \mathrm{Sync})$. Finally, using Proposition 3.3.4, one checks that the unit $\eta \colon X \to \Delta_i' \Sigma_i' X$ is an isomorphism, so Σ_i' is fully faithful. $\qquad\square$

Having introduced and related the toposes of interval sheaves in order to appropriately capture behaviors as sections over lengths of time, we now proceed to the formalization of abstract systems as algebras for wiring diagram operads, typed in those sheaves.

4. Machines as generalizations of dynamical systems

In this chapter, we turn to this work's basic goal: the explicit description of general sorts of processes, called *machines*, in terms of interval sheaves. These machines take on many forms, from discrete and continuous dynamical systems to more general deterministic or total systems, to even more abstract objects. What all these examples have in common is that a collection of machines can be interconnected to form a new machine, and this operation is coherent with respect to nesting of wiring diagrams. In other words, all of our machines will be algebras on an operad of wiring diagrams, as defined in Chapter 2. Subsequently, in Chapter 5 we produce algebra maps that relate all of these various examples.

Each machine will have an interface consisting of input and output ports, where each port carries a sheaf of 'possible signal behaviors', as in

$$A \,\text{—}\!\boxed{\;S\;}\!\text{—}\, B$$
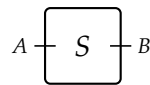
The possible behaviors of the whole machine—including what occurs on its ports—is also represented by a sheaf S. We first give a very general definition: a machine is a span $A \leftarrow S \rightarrow B$. Then we restrict our attention to *total* processes, which have the property that for any initial state and compatible input behavior, there exists a compatible state evolution. We also consider *deterministic* processes, which have the property that the state evolution is unique given any input. Total and deterministic machines are discussed in Section 4.2, and their compositionality is proven in Section 4.3. In the next chapter, we will see that while discrete dynamical systems as defined in Definition 2.3.1 are always both total and deterministic, this is not the case for continuous dynamical systems of Definition 2.3.3 (basically because solutions to general ODEs on unbounded domains may not exist and may not be unique; see Remark 5.1.3).

In Section 4.5 we discuss another related notion, namely that of *contracts* on machines—which may or may not be satisfied by a given machine—that express behavioral guarantees in terms of inputs and outputs. We will show that contracts also form a wiring diagram algebra.

4.1. **Continuous machines.** We begin with the most general notion of process that we will use. It primarily serves the purpose of being all-inclusive, so that the various systems of interest can be seen as special cases. The type of information handled is formalized by continuous interval sheaves, Definition 3.2.2, whereas the algebraic structure is formalized by our preliminary work on span algebras in Section 2.4. While such machines ostensibly have a notion of input and output, the definition is in fact symmetric, so it fits in with the work of [WP13].

Definition 4.1.1. Let $A, B \in \widetilde{\mathbf{Int}}$ be continuous interval sheaves. A *continuous* (A, B)-*machine* is a span

$$
\begin{array}{ccc}
 & S & \\
{}^{p^{\mathrm{i}}}\swarrow & & \searrow{}^{p^{\mathrm{o}}} \\
A & & B
\end{array}
$$
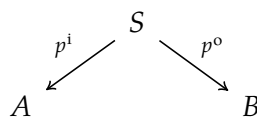
in the topos $\widetilde{\mathbf{Int}}$; equivalently, it is a sheaf S together with a sheaf map $p \colon S \rightarrow A \times B$.

We refer to A as the *input sheaf*, to B as the *output sheaf*, and to S as the *state sheaf*. Their sections of arbitrary lengths ℓ should be thought of as all possible information inputted, worked out and outputted by the machine, during a length of time ℓ. We call p^{i} the *input sheaf map* and p^{o} the *output sheaf map*; they are given by functions $p^{\mathrm{i}}_{\ell} \colon S(\ell) \rightarrow A(\ell)$, $p^{\mathrm{o}}_{\ell} \colon S(\ell) \rightarrow B(\ell)$ on sections of length ℓ. We denote by $\mathsf{Mch}(A, B) \coloneqq \widetilde{\mathbf{Int}}/(A \times B)$, or sometimes by $\mathsf{Mch}_C(A, B)$, the topos of all continuous (A, B)-machines.

Example 4.1.2. Here we collect some examples of continuous machines.

(1) If $f: A \to B$ is a morphism of **Int**-sheaves, there is a continuous (A, B)-machine given by $(\mathrm{id}_A, f): A \to A \times B$.

(2) If A and B are **Int**-sheaves, the identity span $A \times B \to A \times B$ corresponds to the machine for which input and output are completely uncoupled.

(3) For any sheaf A and $\epsilon > 0$, there is a machine (described in Example 4.2.8), which acts as an ϵ-delay: the A input at any time is output after a delay of ϵ-seconds.

(4) In Section 2.3 we discussed discrete and continuous dynamical systems. In Propositions 5.1.1 and 5.1.2, we will give algebra maps realizing them as machines.

Let $\mathcal{W}_{\widetilde{\mathbf{Int}}}$ denote the category of $\widetilde{\mathbf{Int}}$-labeled boxes and wiring diagrams, as in Section 2.2. As in the case of discrete and continuous dynamical systems, the important aspect of these newly-defined machines is that they can be arbitrarily wired together to form new continuous machines, i.e. they are algebras for the wiring diagram operad $\mathcal{W}_{\widetilde{\mathbf{Int}}}$. The following proposition is a corollary of Proposition 2.4.1, for $\mathcal{C} = \widetilde{\mathbf{Int}}$ the finitely complete category of continuous interval sheaves.

Proposition 4.1.3. *Continuous machines form a $\mathcal{W}_{\widetilde{\mathbf{Int}}}$-algebra* $\mathsf{Mch}: \mathcal{W}_{\widetilde{\mathbf{Int}}} \to \mathbf{Cat}$.

Proof. Recall from (4) that to each object (box) $X \in \mathcal{W}_{\widetilde{\mathbf{Int}}}$ we can associate two sheaves, $\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}} \in \widetilde{\mathbf{Int}}$, by taking the product of the types of the input and output $\widetilde{\mathbf{Int}}$-typed sets. The algebra is

$$
\begin{array}{ccc}
\mathsf{Mch}: \mathcal{W}_{\widetilde{\mathbf{Int}}} & \xrightarrow{\hspace{3cm}} & \mathbf{Cat} \\[1em]
(X^{\mathrm{in}}, X^{\mathrm{out}}) & \longmapsto & \widetilde{\mathbf{Int}}/\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \\
\phi \downarrow & & \downarrow \mathsf{Mch}(\phi) \\
(Y^{\mathrm{in}}, Y^{\mathrm{out}}) & \longmapsto & \widetilde{\mathbf{Int}}/\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}
\end{array}
\tag{33}
$$

where $\mathsf{Mch}(\phi)$ maps the span $(p^{\mathrm{i}}, p^{\mathrm{o}}): S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$ to the span $(q^{\mathrm{i}}, q^{\mathrm{o}}): T \to \widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}$ formed by first taking the pullback of $(p^{\mathrm{i}}, p^{\mathrm{o}})$ along $\widehat{\phi^{\mathrm{in}}}$ and then postcomposing with $\widehat{\phi^{\mathrm{out}}}$, as showed in (20). Since limits in $\widetilde{\mathbf{Int}}$ are formed pointwise, we can describe the set of length-ℓ sections of the new state sheaf T explicitly:

$$
T(\ell) \cong \left\{ (s, y) \in S(\ell) \times \widehat{Y^{\mathrm{in}}}(\ell) \mid p_\ell^{\mathrm{i}}(s) = \widehat{\phi_\ell^{\mathrm{in}}}\left(y, p_\ell^{\mathrm{o}}(s)\right) \right\}.
\tag{34}
$$

Roughly, T represents state evolutions of the machine $(S, p^{\mathrm{i}}, p^{\mathrm{o}})$ inhabiting the inside box X, together with ϕ^{in}-compatible sections of Y-input. With this representation, the input sheaf map of the formed machine is $q_\ell^{\mathrm{i}}(s, y) = y$, and the output sheaf map is $q^{\mathrm{o}} = \widehat{\phi^{\mathrm{out}}} \circ p^{\mathrm{o}}$. The symmetric monoidal structure is as in (19). $\qquad\square$

Therefore the above algebra of continuous machines is in fact the algebra of $\widetilde{\mathbf{Int}}$-span systems of Definition 2.4.2. Notice that even though for the topos $\widetilde{\mathbf{Int}}$, the functor $\mathsf{Mch}(\phi)$ (21) has a right adjoint and preserves pullbacks, it does not preserve the terminal object, so it is again the inverse image part of a pre-geometric morphism but not of a geometric morphism.

4.2. **Total, deterministic, and inertial machines.** Our goal in this section is to describe certain subclasses of continuous machines, Definition 4.1.1. Suppose a continuous machine is in a state germ $s_0 \in S(0)$ corresponding to a certain input germ $a_0 := p^i(s_0) \in A(0)$; these can be thought as initial instantaneous values for the state and input of the machine. If a_0 is extended to some longer input behavior $a \in A(\ell)$, e.g. information flows into the machine, s_0 may or may not extend to some state behavior s that accommodates that input a, i.e. with $p^i(s) = a$, meaning that the machine 'runs'. There may be more than one such extension, or none at all.

(35)
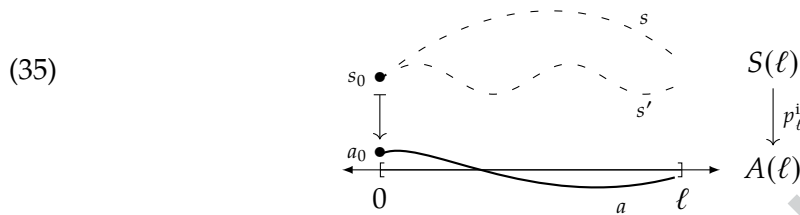
The idea is that a machine is *total* if there is at least one state extension, and it is *deterministic* if there is at most one; their formal description (Definition 4.2.2) will be accomplished by imposing conditions on the input and output sheaf maps. In particular, continuous machines are generally neither total nor deterministic.

Recall the *ϵ-extension* functor $\mathsf{Ext}_\epsilon \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}$ introduced after Definition 3.2.8, where for a sheaf A we have $\mathsf{Ext}_\epsilon A(\ell) = A(\ell + \epsilon)$, as well as the two natural transformations λ and ρ (26) of left and right restriction. For any continuous machine $(p^i, p^o) \colon S \to A \times B$, the pullback

has as elements pairs (a, s_0) of inputs over some length ϵ and state germs in the p^i-fibre of the left restriction $a|_{[0,0]}$. Therefore this already formalizes the initial state germ - input section situation of (35). We employ this to formally express the desired conditions on p^i.

Proposition 4.2.1. *Let $p \colon S \to A$ be a continuous sheaf map. For any $\epsilon \geq 0$, consider the outer naturality square for λ in $\widetilde{\mathbf{Int}}$*

(36)

where h^ϵ is the universal map to the pullback S'. The following are equivalent:

    (1) *for all $\epsilon > 0$, the sheaf map $h^\epsilon \colon \mathsf{Ext}_\epsilon S \to S'$ is an epimorphism (resp. monomorphism)*;

    (2) *for all $\epsilon > 0$, the function $h_0^\epsilon \colon S(\epsilon) \to S'(0)$ is surjective (resp. injective)*;

    (3) *there exists $\delta > 0$ such that for all $0 \leq \delta' \leq \delta$ the sheaf map $h^{\delta'} \colon \mathsf{Ext}_{\delta'} S \to S'$ is an epimorphism (resp. monomorphism);*

    (4) *there exists $\delta > 0$ such that for all $0 \leq \delta' \leq \delta$ the function $h_0^{\delta'} \colon S(\delta') \to S'(0)$ is surjective (resp. injective).*

Proof. When $\epsilon = 0$, we have that $\lambda_A = \mathrm{id}$ and $\lambda_S = \mathrm{id}$, so $h^\epsilon = \mathrm{id}$. Thus it is clear that 1$\Rightarrow$3 and that 2$\Rightarrow$4, e.g. choose $\delta := 1$. By Lemma 3.2.10, the sheaf map h^ϵ is an epi (resp. mono) if and only if for all $\ell \in \mathbf{Int}$ the function h_ℓ^ϵ is epi (resp. mono). In particular, taking $\ell = 0$ we have 1$\Rightarrow$2 and 3$\Rightarrow$4.

    It thus suffices to prove 4$\Rightarrow$1; so assume 4 holds for some $\delta > 0$, and choose $\epsilon > 0$. Any $s' \in S'(\ell)$ can be identified with a pair $(s, a) \in S(\ell) \times A(\ell + \epsilon)$ such that $p(s) = a|_{[0,\ell]}$. For the surjectivity claim, we have that $h_0^{\delta'}$ is surjective for all $0 \leq \delta' \leq \delta$, and the goal is to show h_ℓ^ϵ is surjective; it suffices to find an extension $\bar{s} \in S(t + \epsilon)$ of s over a.

    Since $\mathbb{R}$ is a Euclidean domain, there exists a unique $N \in \mathbb{N}$ and $0 \leq \delta' \leq \delta$ such that $\epsilon = N\delta + \delta'$. Then $(a|_{[\ell,\ell+\delta']}, \rho_0(s)) \in S'(0)$ and since $h_0^{\delta'}$ is surjective, there exists an extension $s_0' \in S(\delta')$ emanating from $\rho_0(s)$ above $a|_{[\ell,\ell+\delta']}$. Gluing this to s we obtain $s_0 := s * s_0' \in S(\ell + \delta')$ extending s over $a_0 := a|_{[0,\ell+\delta']}$.

    If $N = 0$ we are done, so we proceed by induction on N. Given $s_n \in S(\ell + \delta' + n\delta)$, extending s over $a_n := a|_{[0,\ell+\delta'+n\delta]}$ we use the surjectivity of h_0^δ to extend it, exactly as above, to find s_n' emanating from $\lambda_0(s_n)$ over a_n. At the end, we have the desired lift $s_N \in S(\ell + \epsilon)$.

    Assume now that $h_0^{\delta'}$ is injective for all $0 \leq \delta' \leq \delta$. Then there is at most one extension $s_0' \in S(\delta')$ emanating from $\rho_0(s)$ over $a|_{[\ell,\ell+\delta']}$. Thus there is at most one $s_0 \in S(\ell + \delta')$ extending s over $a|_{[0,\ell+\delta']}$. The proof concludes analogously to the surjective case above. $\square$

    A generalization and purely formal proof of the above result is given in Lemma 5.2.7.

Definition 4.2.2. We will say that a sheaf morphism $p \colon S \to A$ is *total* (resp. *deterministic*) if it satisfies the equivalent conditions of Proposition 4.2.1.

Remark 4.2.3. In Appendix A, we discuss an equivalence of categories between $\mathbf{Int}$-sheaves and discrete Conduché fibrations over the monoid $\mathbb{R}_{\geq 0}$. Under that correspondence, a morphism $p \colon S \to A$ in $\widetilde{\mathbf{Int}}$ is both total and deterministic if and only if its associated functor $\overline{p} \colon \overline{S} \to \overline{A}$ is a discrete opfibration; see Appendix A.3.

    It would be reasonable to define a continuous machine $(p^{\mathrm{i}}, p^{\mathrm{o}}) \colon S \to A \times B$ to be total (resp. deterministic) if and only if the input map p^{i} is, as this matches our intuitive notion (35). Indeed, condition 2 says that for every pair $(s_0, a) \in S(0) \times A(\epsilon)$ with $a|_{[0,0]} = p_0^{\mathrm{i}}(s_0)$, there exists some $s \in S(\epsilon)$ above a which extends s_0, for every $\epsilon > 0$. However, such a definition turns out to not be closed under feedback composition. In order for total or deterministic machines to form a $\mathcal{W}$-algebra, we must add an extra condition, this time on their output map. We introduce a notion of *inertia*, in which a machine's current state determines not only its current output, but also a small amount of its future output.

Definition 4.2.4. A sheaf map $p\colon S \to B$ is called ϵ-*inertial* when there exists a factorization

(37)
$$
\begin{array}{ccc}
 & & \mathsf{Ext}_\epsilon B \\
 & \overset{\overline{p}}{\nearrow} & \downarrow{\scriptstyle \lambda} \\
S & \xrightarrow[p]{} & B
\end{array}
$$

through the ϵ-extension of B via a sheaf map $\overline{p}$; it is called *inertial* when it is ϵ-inertial for some $\epsilon > 0$. Explicitly, $\overline{p}_\ell(s)|_{[0,\ell]} = p_\ell(s) \in B(\ell)$ for any section $s \in S(\ell)$.

We say a continuous machine $(p^{\mathrm{i}}, p^{\mathrm{o}})\colon S \to A \times B$ is *inertial* (resp. ϵ-*inertial*) if the output map p^{o} is. Inside $\mathsf{Mch}(A, B)$, the full category of inertial (resp. ϵ-inertial) machines is denoted $\mathsf{Mch}^{\mathsf{in}}(A, B)$ (resp. $\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(A, B)$).

The following expression of inertiality via a lifting in a naturality square is equivalent, and may look more intuitive.

Lemma 4.2.5. *A sheaf map p is ϵ-inertial when there exists a lift $\overline{p}$ as in*

(38)
$$
\begin{array}{ccc}
\mathsf{Ext}_\epsilon S & \xrightarrow{\mathsf{Ext}_\epsilon p} & \mathsf{Ext}_\epsilon B \\
{\scriptstyle \lambda_B}\downarrow & \overset{\overline{p}}{\nearrow} & \downarrow{\scriptstyle \lambda_S} \\
S & \xrightarrow[p]{} & B
\end{array}
$$

Proof. By Definition 4.2.4, it suffices to show that if p is ϵ-inertial then the top triangle commutes. Since $(\mathsf{Ext}_\epsilon p)_\ell = p_{\ell+\epsilon}$ for $\ell \in \mathbf{Int}$, we need to show that the lower triangle in the following square commutes:

$$
\begin{array}{ccc}
 & & B(\ell + 2\epsilon) \\
 & \overset{\overline{p}_{\ell+\epsilon}}{\nearrow} & \downarrow{\scriptstyle \lambda_{\ell+\epsilon}} \\
S(\ell + \epsilon) & \xrightarrow{p_{\ell+\epsilon}} & B(\ell + \epsilon) \\
{\scriptstyle \lambda_\ell}\downarrow & \nearrow{\scriptstyle \overline{p}_\ell} & \\
S(\ell) & &
\end{array}
$$

The top triangle commutes because p is ϵ-inertial, and the outer square commutes because $\overline{p}$ is a sheaf map, therefore has natural components; thus the lower triangle does too. $\quad\square$

Remark 4.2.6. In most cases of interest, if a map $p\colon S \to B$ is inertial then the lift $\overline{p}$ in (37) is unique. Indeed, suppose $\lambda\colon \mathsf{Ext}_\epsilon S \to S$ is surjective, meaning that any short section can be extended in some way; we refer to such sheaves S as *extensible*. In this case, by (38), there is at most one $\overline{p}$ with $\overline{p} \circ \lambda = \mathsf{Ext}_\epsilon p$.

Note that if p is ϵ-inertial and $\delta \leq \epsilon$, then p is δ-inertial as well. Thus, for any pair of sheaves A, B there is a fully faithful functor $\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(A, B) \to \mathsf{Mch}^{\delta\text{-}\mathsf{in}}(A, B)$ given by left restriction, and the colimit of the directed system is $\mathsf{Mch}^{\mathsf{in}}$:

(39)
$$
\mathsf{Mch}^{\mathsf{in}}(A, B) \cong \operatorname*{colim}_{\epsilon > 0} \left(\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(A, B) \right).
$$

We are now ready to define total and deterministic machines, in such a way that they are closed under all wiring diagram operations (Proposition 4.3.2).

Definition 4.2.7. A continuous machine $(p^{\mathrm{i}}, p^{\mathrm{o}})\colon S \to A \times B$ is *total* (resp. *deterministic*) if

- the input map p^{i} is total (resp. deterministic) in the sense of Definition 4.2.2; and
- the output map p^{o} is inertial in the sense of Definition 4.2.4.

We denote the full subcategory of $\mathsf{Mch}(A, B)$ of total, resp. deterministic, (A, B)-machines by $\mathsf{Mch}^{\mathrm{t}}(A, B)$, resp. $\mathsf{Mch}^{\mathrm{d}}(A, B)$. Their intersection $\mathsf{Mch}^{\mathrm{td}}(A, B)$ is the set of machines that are both total and deterministic, i.e. machines for which the sheaf map h^ϵ in (36) is an isomorphism.

Example 4.2.8 (Delay Box)*.* For any sheaf B, we can define a total and deterministic (B, B)-machine $B \, \boxed{D^\epsilon} \, B$ that takes input of type B and delays it for time ϵ: it is the span

$$B \xleftarrow{\rho} \mathsf{Ext}_\epsilon B \xrightarrow{\lambda} B$$

in $\widetilde{\mathbf{Int}}$, where ρ and λ are the right and left restrictions as in (26). It is clear that λ is ϵ-inertial. To see that ρ is both total and deterministic, one checks that the map $h^\epsilon\colon \mathsf{Ext}_{2\epsilon}B \to \mathsf{Ext}_\epsilon B \times_B \mathsf{Ext}_\epsilon B$ in (36) is an isomorphism.

The following examples show that the conditions of totality and determinism on input sheaf maps of machines with non-inertial output maps are indeed not closed under arbitrary machines nesting, as implied earlier.

Example 4.2.9*.* Consider the wiring diagram $\phi\colon X \to 0$, see (8), shown below
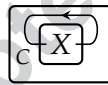
which consists of $(\widehat{\phi^{\mathrm{in}}}\colon \mathbf{1} \times C \xrightarrow{\sim} C, \widehat{\phi^{\mathrm{out}}}\colon C \xrightarrow{!} \mathbf{1})$ in $\widetilde{\mathbf{Int}}$ as in (14), where $\mathbf{1}$ is the terminal sheaf. Any inhabitant continuous machine $P \coloneqq (p^{\mathrm{i}}, p^{\mathrm{o}})\colon S \to C \times C$ becomes, via (20), $\mathsf{Mch}(\phi)(P) = T \to \mathbf{1} \times \mathbf{1}$ with new sheaf of states given by $T(\ell) = \{s \in S(\ell) \mid p^{\mathrm{i}}_\ell(s) = p^{\mathrm{o}}_\ell(s)\}$ as in (34), namely the collection of all state sections that have the same input and output.

Take $C = \mathsf{Fnc}(\underline{2})$ to be the sheaf of functions on the set $\underline{2} \coloneqq \{1, 2\}$ as described in Example 3.2.6(2), given by $C(\ell) = \{f\colon [0, \ell] \to \{1, 2\}\}$. The machine $(\mathrm{id}_C, \mathrm{id}_C)\colon C \to C \times C$ is such that its input map is deterministic (and total); indeed, given any input function $f\colon [0, \ell] \to \{1, 2\}$ and an initial state germ above it, namely $f(0)$, then there exists a unique state extension mapping to f via p^{i}, namely f itself. However, its composite $\mathsf{Mch}(\phi)(P)$ has the same state sheaf C since all functions have the same input and output via the identities, whereas $C \to \mathbf{1} \times \mathbf{1}$ does not have a deterministic input anymore: there are multiple functions that only agree on 0.

Now take $C = \mathsf{Yon}_1$ to be the representable sheaf on the interval $1 = [0, 1]$ as described in Example 3.2.6(1), given by $\mathsf{Yon}_1(\ell) = \mathbf{Int}(\ell, 1)$. Notice that for lengths $\ell > 1$, these sets are empty. Again, the machine $(\mathrm{id}_C, \mathrm{id}_C)\colon C \to C \times C$ has total (and deterministic), however its composite $C \to \mathbf{1} \times \mathbf{1}$ does not have a total input map anymore: the diagram (36) would require for example $\lambda\colon \mathsf{Ext}_\epsilon(C)(1) = C(1 + \epsilon) \to C(1)$ to be surjective, when $C(1 + \epsilon) = \emptyset$.

Notice that in both cases above, the output map id_C was not inertial. For example, suppose that there exists a factorization $C \xrightarrow{\ \overline{p}\ } \mathsf{Ext}_\epsilon(C) \xrightarrow{\ \lambda\ } C$ of the identity sheaf morphism of the first machine. Considering the components $\overline{p}_1$ and $\overline{p}_2$ for $\epsilon = 1$, naturality

implies the commutativity of

$$
\begin{array}{ccc}
C(2) & \xrightarrow{\lambda_1} & C(1) \\
\bar{p}_2 \downarrow & & \downarrow \bar{p}_1 \\
C(3) & \xrightarrow{\lambda_2} & C(2)
\end{array}
$$

where the lower composite is the identity on $C(2)$; that would imply that the left restriction λ is injective, which is clearly false.

4.3. **Closure under feedback.** For the remainder of this section, we write $\mathcal{W}$ to denote the symmetric monoidal category $\mathcal{W}_{\widetilde{\mathbf{Int}}}$ of $\widetilde{\mathbf{Int}}$-labeled boxes and wiring diagrams. We saw in Proposition 4.1.3 that continuous machines form a $\mathcal{W}$-algebra $\mathsf{Mch} \colon \mathcal{W} \to \mathbf{Cat}$; now we show that inertial (Definition 4.2.4), as well as total machines and deterministic machines (Definition 4.2.7) do too.

Proposition 4.3.1. *Inertial continuous machines* (*resp. ϵ-inertial machines, for any $\epsilon > 0$*) *form a subalgebra* $\mathsf{Mch}^{\mathsf{in}} \colon \mathcal{W} \to \mathbf{Cat}$ (*resp.* $\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}$). *Moreover, we have a colimit of algebras*

$$
\mathsf{Mch}^{\mathsf{in}} \cong \operatorname*{colim}_{\epsilon > 0}(\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}).
$$

Proof. Given a wiring diagram morphism $\phi \colon X \to Y$, we restrict (33) to obtain a functor

$$
\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(X) \subseteq \mathsf{Mch}(X) \xrightarrow{\ \mathsf{Mch}(\phi)\ } \mathsf{Mch}(Y);
$$

we will show that it factors through the full subcategory $\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(Y)$, for the same $\epsilon > 0$. Suppose that $p = (p^{\mathsf{i}}, p^{\mathsf{o}}) \colon S \to \widehat{X^{\mathsf{in}}} \times \widehat{X^{\mathsf{out}}}$ is a machine and that p^{o} factors as $\lambda \circ \bar{p}$ (37). This is shown in the upper-middle square below

$$(40)$$

The construction for $\mathsf{Mch}(\phi)(S, p) = (T, q)$ as in (20) is the left-hand pullback, along with the bottom-left horizontal morphism, thus defining $q = (q^{\mathsf{i}}, q^{\mathsf{o}}) \colon T \to \widehat{Y^{\mathsf{in}}} \times \widehat{Y^{\mathsf{out}}}$. The bottom (trapezoid-shaped) sub-diagram trivially commutes, and the right-hand diagram commutes because λ is a natural transformation. Thus the outer square commutes, exhibiting a factorization of q^{o} through $\mathsf{Ext}_\epsilon(\widehat{Y^{\mathsf{out}}})$. Hence we do have a subfunctor $\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}} \colon \mathcal{W} \to \mathbf{Cat}$.

If $0 < \delta \leq \epsilon$, the following diagram of categories commutes

$$
\begin{array}{ccc}
\mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(X) & \xrightarrow{\ \mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(\phi)\ } & \mathsf{Mch}^{\epsilon\text{-}\mathsf{in}}(Y) \\
\downarrow & & \downarrow \\
\mathsf{Mch}^{\delta\text{-}\mathsf{in}}(X) & \xrightarrow{\ \mathsf{Mch}^{\delta\text{-}\mathsf{in}}(\phi)\ } & \mathsf{Mch}^{\delta\text{-}\mathsf{in}}(Y)
\end{array}
$$

so we have a natural transformation $\mathsf{Mch}^{\epsilon\text{-in}} \Rightarrow \mathsf{Mch}^{\delta\text{-in}}$, and the colimit of the directed system is Mch^{in} by (39). It remains to show that all these functors $\mathcal{W} \to \mathbf{Cat}$, and maps between them, are monoidal.

It is easy to see that the product (19) of ϵ-inertial machines is ϵ-inertial, because Ext_ϵ preserves products of sheaves, see Lemma 3.2.9. Thus $\mathsf{Mch}^{\epsilon\text{-in}}$ is indeed a $\mathcal{W}$-algebra, and moreover the natural transformation $\mathsf{Mch}^{\epsilon\text{-in}} \Rightarrow \mathsf{Mch}^{\delta\text{-in}}$ described above is clearly monoidal. We still need to define a monoidal structure on the colimit Mch^{in}: given output sheaf maps that factorize through $S \to \mathsf{Ext}_{\epsilon_1}(\widehat{X^{\text{out}}})$ and $P \to \mathsf{Ext}_{\epsilon_2}(\widehat{Z^{\text{out}}})$ for $\epsilon_1, \epsilon_2 > 0$, let $\epsilon := \min(\epsilon_1, \epsilon_2)$. Then by left restriction (if necessary), the product output sheaf map factorizes through $S \times P \to \mathsf{Ext}_\epsilon(\widehat{X^{\text{out}}} \times \widehat{Z^{\text{out}}})$ therefore is also inertial for this new $\epsilon > 0$. $\square$

Proposition 4.3.2. *Total machines form a subalgebra* $\mathsf{Mch}^{\text{t}} \colon \mathcal{W} \to \mathbf{Cat}$*. Similarly, deterministic machines form a subalgebra* $\mathsf{Mch}^{\text{d}} \colon \mathcal{W} \to \mathbf{Cat}$*. Their intersection is a subalgebra* $\mathsf{Mch}^{\text{td}} :=$ $\mathsf{Mch}^{\text{t}} \cap \mathsf{Mch}^{\text{d}}$*.*

Proof. Given an object $X = (X^{\text{in}}, X^{\text{out}})$, define the full subcategories $\mathsf{Mch}^{\text{t}}(X), \mathsf{Mch}^{\text{d}}(X)$ of $\mathsf{Mch}(\widehat{X^{\text{in}}}, \widehat{X^{\text{out}}})$ as in Definition 4.2.7. Given a morphism $\phi \colon X \to Y$, we restrict to obtain functors

$$\mathsf{Mch}^{\text{t}}(X) \subseteq \mathsf{Mch}(X) \xrightarrow{\mathsf{Mch}(\phi)} \mathsf{Mch}(Y) \qquad \text{and} \qquad \mathsf{Mch}^{\text{d}}(X) \subseteq \mathsf{Mch}(X) \xrightarrow{\mathsf{Mch}(\phi)} \mathsf{Mch}(Y)$$

described in Proposition 4.1.3; we need to show that they factor through the respective full subcategories $\mathsf{Mch}^{\text{t}}(Y)$ and $\mathsf{Mch}^{\text{d}}(Y)$, and the result for Mch^{td} follows trivially.

A total (resp. deterministic) machine $p = (p^{\text{i}}, p^{\text{o}}) \colon S \to \widehat{X^{\text{in}}} \times \widehat{X^{\text{out}}}$ is ϵ-inertial for some $\epsilon > 0$, meaning that p^{o} is a composite $\lambda \circ \overline{p} \colon S \to \mathsf{Ext}_\epsilon \widehat{X^{\text{out}}} \to \widehat{X^{\text{out}}}$ (37). By Proposition 4.3.1, its image $q = (q^{\text{i}}, q^{\text{o}}) \colon T \to \widehat{Y^{\text{in}}} \times \widehat{Y^{\text{out}}}$ under $\mathsf{Mch}(\phi)$ is also ϵ-inertial for the same ϵ, so $q^{\text{o}} = \lambda \circ \overline{q}$ for some $\overline{q}$ as exhibited by (40). In particular, from the top-right part of that diagram we deduce that this factorization coincides with one through $\widehat{X^{\text{out}}}$, namely

$$(41) \qquad q^{\text{o}} = T \xrightarrow{k} S \xrightarrow{\overline{p}} \mathsf{Ext}_\epsilon \widehat{X^{\text{out}}} \xrightarrow{\lambda} \widehat{X^{\text{out}}} \xrightarrow{\phi^{\text{out}}} \widehat{Y^{\text{out}}}.$$

It suffices to check that the resulting input sheaf map q^{i} is total (resp. deterministic). We use the formulation in Proposition 4.2.1(3); that is, we will show that the sheaf map $h^{\epsilon'}$ in (36) is an epimorphism (resp. monomorphism) for arbitrary $\epsilon' \leq \epsilon$. But note that if q^{o} is ϵ-inertial then it is also ϵ'-inertial, so we may take $\epsilon' = \epsilon$. First of all, the bottom squares below are pullbacks in a straightforward way; we define S' and T' to be the top pullbacks (42)

Note that the map p' on the left is the one defined in Proposition 4.2.1, and similarly is q' if we observe that the top pullbacks are along identities with respect to the second components. Therefore the outer pullbacks coincide with the ones in (36) for the machines S and T. Now consider the diagram

where the two sides are precisely the pullbacks of (42). Notice how the two front vertical composites describe the ϵ-inertial machine $S \to \widehat{X^{\text{in}}} \times \widehat{X^{\text{out}}}$ on the right, and its resulting (also ϵ-inertial) machine $T \to \widehat{Y^{\text{in}}} \times \widehat{Y^{\text{out}}}$ on the left by (41). Now the front rectangle is a pullback by definition (20) of machine composition, and one can check that the lower bottom square is also a pullback. Therefore the back face is a pullback, which we re-write as the bottom square below:

(43)

Since Ext_ϵ preserves limits by Lemma 3.2.9, the big rectangle is a pullback, so the top is too. This diagram takes place in a topos—in particular, a regular category—so if h^ϵ is epi (resp. mono) then i^ϵ is too. Therefore if p is a total (resp. deterministic) machine, so is $\text{Mch}(\phi)(p)$.

Finally, it is easy to check that if $p^{\text{i}} \colon P \to \widehat{X^{\text{in}}}$ and $q^{\text{i}} \colon T \to \widehat{Z^{\text{in}}}$ are total (resp. deterministic), their product $p^{\text{i}} \times q^{\text{i}}$ is too, so lax monoidality (19) follows. □

Notably, since the above conditions are only sufficient and not necessary — the pullback (43) may produce an epimorphism or monomorphism i^ϵ on the left even if the initial h^ϵ on the right is not — it could be the case that a composite machine is total and deterministic without its sub-components being so. Even more, the two cases of Example 4.2.9 coincidentally maintain their totality and determinism respectively, despite not being inertial. In this work, our goal was to establish sufficient conditions for totality and determinism to be

carried over from subsystems to their composite, and this is what was accomplished in Proposition 4.3.2.

4.4. **Discrete and synchronous variations.** One of the central purposes of the current work is to allow discrete-time and continuous-time systems to be incorporated within the same framework. In this section, we define discrete analogues of the machines described so far; all the results go through with nearly identical proofs, so we only provide the definitions and statements. In the following Chapter 5, we compare the discrete and continuous machines via wiring diagram algebra maps.

Discrete machines have $\mathbf{Int}_N$-sheaves, rather than $\mathbf{Int}$-sheaves, labeling their input and output ports; see Section 3.2 were continuous and discrete interval sheaves were introduced. All of the ideas of Sections 4.2 and 4.3 can be modified to this setting, with the main difference being that the positive real $\epsilon \in \mathbb{R}_{\geq 0}$ used in the various extension sheaves (Definition 3.2.8) is here replaced by the positive integer $1 \in \mathbb{N}$. For example, compare the following to Proposition 4.2.1 and Definitions 4.2.4 and 4.2.7.

Definition 4.4.1. Let $p\colon S \to A$ be a discrete sheaf map. Consider the outer naturality square for λ in $\widetilde{\mathbf{Int}_N}$

$$
\begin{array}{ccc}
\mathsf{Ext}_1 S & \xrightarrow{\ \lambda\ } & \\
\ \ \downarrow{\scriptstyle h^1} & & \\
& S' \longrightarrow & S \\
\mathsf{Ext}_1 p\ \downarrow & {\scriptstyle p'}\downarrow\ \ \lrcorner & \downarrow{\scriptstyle p} \\
& \mathsf{Ext}_1 A \xrightarrow{\ \lambda\ } & A
\end{array}
$$
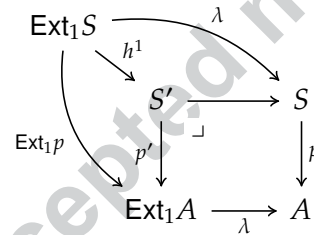
where $h_1 \colon \mathsf{Ext}_1 S \to S'$ is the universal map to the pullback S'. Then p is *total* (resp. *deterministic*) if h_1 is an epimorphism (resp. monomorphism); p is *inertial* if it factors through $\lambda \colon \mathsf{Ext}_1 A \to A$.

A *discrete machine* is a span $p = (p^{\mathrm{i}}, p^{\mathrm{o}})\colon S \to A \times B$ in $\widetilde{\mathbf{Int}_N}$. The machine is *inertial* if the output map p^{o} is. The machine *total* (resp. *deterministic*) if both the output map p^{o} is inertial and the input map p^{i} is *total* (resp. *deterministic*).

Remark 4.4.2. By an argument similar to that in Proposition 4.2.1, we can simplify Definition 4.4.1 a great deal. Recall from Proposition 3.2.3 that $p\colon S \to A$ can be identified with a graph homomorphism; write $S = (\ S_1 \overset{\mathrm{src}}{\underset{\mathrm{tgt}}{\rightrightarrows}} S_0\ )$ where we let $S_n \coloneqq S(n)$ denote the set of length-n sections of S, and similarly for A. Consider the diagram of sets

(44)
$$
\begin{array}{ccc}
S_1 & \xrightarrow{\ \mathrm{src}\ } & \\
\ \ \downarrow{\scriptstyle h} & & \\
& S' \longrightarrow & S_0 \\
{\scriptstyle p_1}\downarrow & \downarrow\ \ \lrcorner & \downarrow{\scriptstyle p_0} \\
& C_1 \xrightarrow{\ \mathrm{src}\ } & C_0
\end{array}
$$
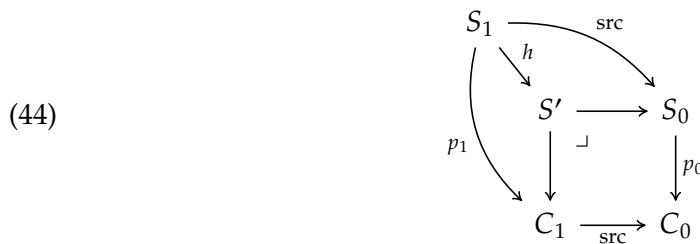
where S' is the pullback and h is the induced function. Then p is total (resp. deterministic) if and only if h is surjective (resp. injective). Notice that $h = h_0^1$ from above.

The proof of Proposition 4.4.3 follows that of Propositions 4.1.3, 4.3.1 and 4.3.2. In particular, $\mathsf{Mch}_N \colon \mathcal{W}_{\widetilde{\mathbf{Int}}_N} \to \mathbf{Cat}$ is the algebra of $\widetilde{\mathbf{Int}}_N$-span systems of Definition 2.4.2.

Proposition 4.4.3. *Discrete machines form a* $\mathcal{W}_{\widetilde{\mathbf{Int}}_N}$*-algebra* $\mathsf{Mch}_N \colon \mathcal{W}_{\widetilde{\mathbf{Int}}_N} \to \mathbf{Cat}$*. Inertial, total, deterministic discrete machines form subalgebras* $\mathsf{Mch}_N^{\mathsf{in}}$, $\mathsf{Mch}_N^{\mathsf{t}}$, $\mathsf{Mch}_N^{\mathsf{d}}$ *(and* $\mathsf{Mch}_N^{\mathsf{td}}$*).*

To compare discrete and continuous systems in Chapter 5, it will be useful to have a mediating construct that handles 'synchronization' in continuous systems. Recall the synchronizing sheaf $\mathsf{Sync} \in \widetilde{\mathbf{Int}}$ (25) and the synchronous sheaves $X \to \mathsf{Sync}$ from Section 3.3. Given an object (resp. map, span) in $\widetilde{\mathbf{Int}}/\mathsf{Sync}$, we refer to its image under the forgetful $\widetilde{\mathbf{Int}}/\mathsf{Sync} \to \widetilde{\mathbf{Int}}$ as the *underlying* object (resp. map, span) in $\widetilde{\mathbf{Int}}$.

Definition 4.4.4. A morphism $p \colon S \to A$ in $\widetilde{\mathbf{Int}}/\mathsf{Sync}$ is *inertial* (resp. *total, deterministic*) if the underlying map in $\widetilde{\mathbf{Int}}$ is.

A *synchronous machine* is a span $p = (p^{\mathsf{i}}, p^{\mathsf{o}}) \colon S \to A \times B$ in $\widetilde{\mathbf{Int}}/\mathsf{Sync}$. It is *inertial* (resp. *total, deterministic*) if the underlying continuous machine p in $\widetilde{\mathbf{Int}}$ is.

Again, the proof of Proposition 4.4.5 is similar to the analogous results proven above, and in particular $\mathsf{Mch}_{\mathsf{Sync}} \colon \mathcal{W}_{\widetilde{\mathbf{Int}}/\mathsf{Sync}} \to \mathbf{Cat}$ is the algebra of $\widetilde{\mathbf{Int}}/\mathsf{Sync}$-span systems of Definition 2.4.2.

Proposition 4.4.5. *Synchronous machines form a* $\mathcal{W}_{\widetilde{\mathbf{Int}}/\mathsf{Sync}}$*-algebra* $\mathsf{Mch}_{\mathsf{Sync}} \colon \mathcal{W}_{\widetilde{\mathbf{Int}}/\mathsf{Sync}} \to \mathbf{Cat}$*. Inertial, total, deterministic synchronous machines form subalgebras* $\mathsf{Mch}_{\mathsf{Sync}}^{\mathsf{in}}$, $\mathsf{Mch}_{\mathsf{Sync}}^{\mathsf{t}}$, $\mathsf{Mch}_{\mathsf{Sync}}^{\mathsf{d}}$ *(and* $\mathsf{Mch}_{\mathsf{Sync}}^{\mathsf{td}}$*).*

Evidently, continuous machines receive input information continuously through time, discrete machines receive information in discrete 'ticks' $0, 1, 2, ..$ of a global clock, whereas synchronous machines also have a continuous input flow, this time together with an assigned phase $\theta \in [0, 1)$. In Corollary 5.2.1 we will show that there are algebra morphisms realizing any discrete or continuous machine as a synchronous machine.

4.5. **Safety contracts.** We conclude this chapter by a short discussion on *contracts* that we can impose on machines. Intuitively, a contract for a machine should be a set of logical formulas that dictate what sort of input/output behavior is *valid* for it. If the machine inhabits some labeled box X, its contract should be expressed relatively to the sections of the input and output sheaves $\widehat{X^{\mathsf{in}}}, \widehat{X^{\mathsf{out}}}$.

For example, suppose we have a box X and consider the following natural language contract for a discrete machine inhabiting X: "if I ever receive two True's in a row, I will output a False within 5 seconds". For any input/output pair $(i, j) \in \widehat{X^{\mathsf{in}}}(n) \times \widehat{X^{\mathsf{out}}}(n)$, we can evaluate whether the machine satisfies the contract, at least on the interval $[0, 6]$: if the first two inputs are True, $i|_{[0,1]} = \langle \mathsf{True}, \mathsf{True} \rangle$, then the last five outputs $j|_{[2,6]}$ should include a False. For any section s of longer length $n \geq 6$, we say the machine satisfies the contract if it does on every subinterval $[a, a + 6] \subseteq [0, n]$.

But on shorter sections, it is unclear what to do: should one say the contract is satisfied on short intervals or not? One choice is what is sometimes called a *safety contract*. These have the property that if a section validates the contract, then so does any restriction of it. Thus we formalize safety contracts on X as sub-presheaves $C \subseteq U(\widehat{X^{\mathsf{in}}} \times \widehat{X^{\mathsf{out}}}) \cong U\widehat{X^{\mathsf{in}}} \times U\widehat{X^{\mathsf{out}}}$

where U is the forgetful functor into presheaves; namely if a section $x = (i, j)$ is in C then it is valid.[8] Thus we arrive at the following definition.

Definition 4.5.1. Let $A, B \in \widetilde{\mathbf{Int}}$ be continuous interval sheaves, and $U \colon \widetilde{\mathbf{Int}} \to \mathsf{Psh}(\mathbf{Int})$ the forgetful functor. A *safety contract* is a sub-presheaf $C \subseteq UA \times UB$. We say a section $(a, b) \in A \times B$ *validates* the contract if $(a, b) \in C$.

Recall that images exist in any topos, in particular in $\mathsf{Psh}(\mathbf{Int})$. If $p = (p^{\mathrm{i}}, p^{\mathrm{o}}) \colon S \to A \times B$ is a continuous machine as in Definition 4.1.1, we say that it *validates* the contract, denoted $p \models C$, if the image $\mathsf{im}(Up)$ is contained in C, or equivalently Up factors through it:

$$
\begin{array}{ccc}
 & & C \\
 & \nearrow & \downarrow \\
US & \xrightarrow[Up]{} & UA \times UB
\end{array}
$$

In other words, p validates C if, for every section $s \in S$, the associated input and output validates the contract as above, $\big(p^{\mathrm{i}}(s), p^{\mathrm{o}}(s)\big) \in C$.

The collection of all (A, B)-contracts is a poset—in fact a Heyting algebra; it is denoted

$$(45) \qquad \mathsf{Cntr}(A, B) := \mathsf{SubPsh}(A \times B) \equiv \mathsf{Sub}_{\mathsf{Psh}(\mathbf{Int})}(UA \times UB).$$

In what follows, we often elide the forgetful functor U so we may write $C \subseteq A \times B$ to denote $C \subseteq UA \times UB$; in diagrams involving both sheaves and presheaves, everything should be regarded as a presheaf.

Remark 4.5.2. As mentioned at the beginning of this section, contracts should really be written in a logical formalism. What we call safety contracts in Definition 4.5.1 are a reasonable semantics for this formalism; a related logical formalization was later worked out in [SS17].

Proposition 4.5.3. *Safety contracts form a $\mathcal{W}_{\widetilde{\mathbf{Int}}}$-algebra.*

Proof. The proof proceeds like the ones for machines or span-like systems earlier, e.g. Proposition 4.1.3. To any box $X \in \mathcal{W}_{\widetilde{\mathbf{Int}}}$, we associate the posetal category $\mathsf{Cntr}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$ of (45). Given a wiring diagram $\phi \colon X \to Y$ and a contract $C \subseteq \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$, we form $D := \mathsf{Cntr}(\phi)(C)$

$$(46)$$

$$
\begin{array}{ccc}
C' & \longrightarrow & C \\
\downarrow \quad \lrcorner & & \downarrow \\
\widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} & \xrightarrow{(\widehat{\phi^{\mathrm{in}}}, \pi_2)} & \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}} \\
\downarrow {\scriptstyle 1 \times \widehat{\phi^{\mathrm{out}}}} & & \\
\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}} & &
\end{array}
$$

with D mapping from C' and to $\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}$.

---

[8] Using sheaves, rather than presheaves, for safety contracts does not have the intended semantics, because the sheaf condition would imply that the concatenation of any two valid behaviors is valid. It thus effectively disallows historical context from being a consideration in validity. In the example above, any section of length 4 is valid, but some sections of length 8 are not so the gluing of two valid sections is not always valid.

as the subpresheaf of $\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}$ which the image of the pullback C'. Analogously to (21), if $(\exists_f \dashv f^*): \mathsf{Sub}(B) \to \mathsf{Sub}(A)$ are the induced adjunctions for any $f: A \to B$ in the topos $\mathsf{Psh}(\mathbf{Int})$, the functor $\mathsf{Cntr}(\phi)$ is the composite

$$\mathsf{SubPsh}(\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) \xrightarrow{(\widehat{\phi^{\mathrm{in}}}, \pi_2)^*} \mathsf{SubPsh}(\widehat{Y^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) \xrightarrow{\exists_{(1 \times \widehat{\phi^{\mathrm{out}}})}} \mathsf{SubPsh}(\widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}).$$

Functoriality and unitality follow. The lax monoidal structure is again given by the cartesian product like (19), i.e. natural maps $\mathsf{SubPsh}(X) \times \mathsf{SubPsh}(Z) \to \mathsf{SubPsh}(X \times Z)$. □

Finally, we may consider the inhabitant of a wiring diagram box to be a machine with an associated contract, namely any contract validated by it. These also form an algebra, essentially combining Propositions 4.1.3 and 4.5.3; the composed inhabitant is the composite machine, validating the associated composite contract.

Definition 4.5.4. Let $A, B \in \widetilde{\mathbf{Int}}$ be sheaves. An (A, B)-*contracted machine* is a pair $(S, C) \in \mathsf{Mch}(A, B) \times \mathsf{Cntr}(A, B)$ such that $S \models C$.

Contracted machines form a full subcategory $\mathsf{CM}(A, B)$ of $(\mathsf{Mch} \times \mathsf{Cntr})(A, B)$.

Proposition 4.5.5. *Contracted machines form a $\mathcal{W}_{\widetilde{\mathbf{Int}}}$-subalgebra of $\mathsf{Mch} \times \mathsf{Cntr}$.*

Proof. For any $X = (X^{\mathrm{in}}, X^{\mathrm{out}})$, define

$$\mathsf{CM}(X) := \mathsf{CM}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \subseteq \mathsf{Mch}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) \times \mathsf{Cntr}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}).$$

Suppose $\phi: X \to Y$ is a wiring diagram. Given a contracted machine $(p: S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}, m: C \rightarrowtail U\widehat{X^{\mathrm{in}}} \times U\widehat{X^{\mathrm{out}}}) \in \mathsf{CM}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$, applying $\mathsf{Mch}(\phi) \times \mathsf{Cntr}(\phi)$ produces a pair $(q: T \to \widehat{Y^{\mathrm{in}}} \times \widehat{Y^{\mathrm{out}}}, n: D \to U\widehat{Y^{\mathrm{in}}} \times U\widehat{Y^{\mathrm{out}}})$ given by (20) and (46). The following diagram verifies that q validates n:
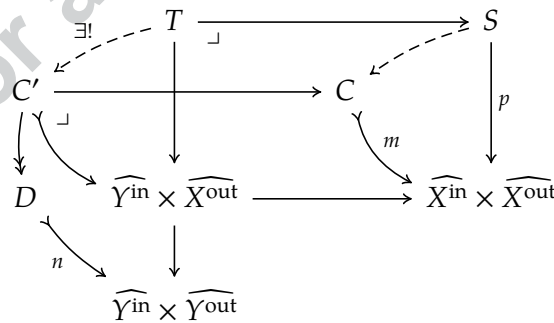
The left hand side composite is precisely the new machine q, due to the upper-side pullback formed; clearly it factors through n. Monoidality is inherited from $\mathsf{Mch} \times \mathsf{Cntr}$, essentially given by the cartesian product. □

We could of course replace Mch with any other version of a machine (total, deterministic and discrete, synchronous) as described in Sections 4.2 and 4.4, and express safety contracts on those. The significance of Proposition 4.5.5 can be summarized as follows: if we arbitrarily interconnect machines which validate specific contracts, the composite machine they form is forced to satisfy their composite contract, which can be specified using the wiring diagram algebraic operations. In other words, we could reason about the

expected valid behavior of the total system only by looking at valid behaviors, and not even state specifications, of the component subsystems.

5. Maps between dynamical systems and machines

In this final chapter of the main text, we describe wiring diagram algebra maps, i.e. monoidal natural transformations (13), between the various algebras we have considered so far. More specifically, we are interested in maps from discrete and continuous dynamical systems of Section 2.3 to the machines considered in Chapter 4, as well as maps between the different kinds of machines (continuous, discrete, etc.) and contracts. These maps, in fact embeddings, allow us to translate one sort of algebra to another, while ensuring consistency of serial, parallel, and feedback composition.

Let us briefly elaborate on the previous sentence for two $\mathcal{W}$-algebras, i.e. lax monoidal functors $F, G \colon \mathcal{W} \to \mathbf{Cat}$. Recall that, given F-inhabitants of boxes $f_i \in F(X_i)$ and a wiring diagram $\phi \colon X_1, \ldots, X_n \to Y$ like (11), we obtain an inhabitant $F(\phi)(f_1, \ldots, f_n) \in F(Y)$ of the outer box via (12). Given a monoidal natural transformation $\alpha \colon F \Rightarrow G$, its components $\alpha_{X_i} \colon F(X_i) \to G(X_i)$ map F-inhabitants to G-inhabitants. The axioms dictate the commutativity of

$$
\begin{array}{ccccc}
F(X_1) \times \ldots \times F(X_n) & \xrightarrow{F_{X_1,\ldots,X_n}} & F(X_1 + \ldots + X_n) & \xrightarrow{F(\phi)} & F(Y) \\
\Big\downarrow{\scriptstyle \alpha_{X_1} \times \ldots \times \alpha_{X_n}} & & \Big\downarrow{\scriptstyle \alpha_{X_1+\ldots+X_n}} & & \Big\downarrow{\scriptstyle \alpha_Y} \\
G(X_1) \times \ldots \times G(X_n) & \xrightarrow[G_{X_1,\ldots,X_n}]{} & G(X_1 + \ldots + X_n) & \xrightarrow[G(\phi)]{} & G(Y)
\end{array}
$$

which explicitly means that whether we first compose the interior systems and then translate via α (upper composite), or first translate and then compose (lower composite), the resulting systems are the same.

We explain in Section 5.1 how to translate from various sorts of dynamical systems to machines, and in Section 5.2 how to translate between various sorts of machines. Specifically, Corollary 5.2.1 establishes synchronous machines $\mathsf{Mch}_{\mathsf{Sync}}$ as the common framework where discrete and continuous machines can both be mapped.

5.1. **Realizing dynamical systems as machines.** Understanding how the motivating examples of discrete and continuous dynamical systems of Section 2.3 fit into our generalized framework of machines, makes the latter much more concrete and accomplishes one of the main goals of this work.

To begin with, we realize discrete dynamical systems (Definition 2.3.1) as discrete machines (Definition 4.4.1), in fact total and deterministic ones. The former are **Set**-based—the input and output signals are elements of a set, as are the states—so the first step is to compare the typing categories of the respective wiring diagram algebras, namely sets with $\mathbf{Int}_N$-sheaves. In Example 3.2.5, for any set S we defined the $\widetilde{\mathbf{Int}}_N$-sheaf $K(S)$, with length-n sections $K(S)(n) = S^{n+1} = \mathrm{Hom}_{\mathbf{Set}}(\{0, \ldots, n\}, S)$, i.e. lists of length $n + 1$ in S. This mapping extends to a (finite-limit preserving) functor

$$(47) \qquad\qquad\qquad\qquad K \colon \mathbf{Set} \to \widetilde{\mathbf{Int}}_N$$

therefore by (9), there is an induced strong monoidal functor $\mathcal{W}_K \colon \mathcal{W}_{\mathbf{Set}} \to \mathcal{W}_{\widetilde{\mathbf{Int}}_N}$ which sends a pair of **Set**-typed finite sets ($\tau \colon X^{\mathrm{in}} \to \mathbf{Set}$, $\tau' \colon X^{\mathrm{out}} \to \mathbf{Set}$) to the $\widetilde{\mathbf{Int}}_N$-typed finite sets ($K \circ \tau \colon X^{\mathrm{in}} \to \widetilde{\mathbf{Int}}_N$, $K \circ \tau' \colon X^{\mathrm{out}} \to \widetilde{\mathbf{Int}}_N$).

Recall from Proposition 4.4.3 the algebra $\mathsf{Mch}^{\mathrm{td}}_N \colon \mathcal{W}_{\widetilde{\mathbf{Int}}_N} \to \mathbf{Cat}$ of total and deterministic discrete machines, and from Proposition 2.3.2 the algebra $\mathsf{DDS} \colon \mathcal{W}_{\mathbf{Set}} \to \mathbf{Cat}$ of discrete dynamical systems.

Proposition 5.1.1. *There exists a morphism of wiring diagram algebras, i.e. monoidal natural transformation*

Proof. We first define the component functor $\beta_X \colon \mathsf{DDS}(X) \to \mathsf{Mch}^{\mathrm{td}}_N(\mathcal{W}_K(X))$ for each box $X = (X^{\mathrm{in}}, X^{\mathrm{out}}) \in \mathcal{W}_{\mathbf{Set}}$, mapping a discrete dynamical system on X to a total and deterministic discrete machine on $\mathcal{W}_K(X)$. We then we check that the β_X's satisfy the necessary naturality and monoidality conditions.

Recall that a discrete dynamical system on a **Set**-labeled box X is a tuple $F = (S, f^{\mathrm{upd}}, f^{\mathrm{rdt}})$, where S is a set, and $f^{\mathrm{upd}} \colon \widehat{X^{\mathrm{in}}} \times S \to S$ and $f^{\mathrm{rdt}} \colon S \to \widehat{X^{\mathrm{out}}}$ are functions. For simplicity, denote by $\mathcal{X}^{\mathrm{in}} \coloneqq K(\widehat{X^{\mathrm{in}}})$ and $\mathcal{X}^{\mathrm{out}} \coloneqq K(\widehat{X^{\mathrm{out}}})$ the image of the input and output sets under K (47), namely the discrete interval sheaves with $n + 1$-element lists as sections of length n.

For each discrete dynamical system F, we can construct a discrete interval sheaf $\mathcal{S}$ by

$$(48) \qquad \mathcal{S}(n) \coloneqq \left\{ (x, s) \colon \{0, \dots, n\} \to \widehat{X^{\mathrm{in}}} \times S \;\middle|\; s_{i+1} = f^{\mathrm{upd}}(x_i, s_i), \quad 0 \le i < n \right\}$$

i.e. finite lists of all pairs of input and state elements, in the order processed by the dynamical system: the $(i+1)^{\mathrm{th}}$ state is determined by the ith state and input. We can now define $\beta_X(F)$ to be the span $(p^{\mathrm{i}}, p^{\mathrm{o}}) \colon \mathcal{S} \to \mathcal{X}^{\mathrm{in}} \times \mathcal{X}^{\mathrm{out}}$ in $\widetilde{\mathbf{Int}}_N$, where $p^{\mathrm{i}}(x, s) = x$ and $p^{\mathrm{o}}(x, s) = f^{\mathrm{rdt}}(s)$; this is the discrete machine that corresponds to F. Notice that if $\beta_X(F) = \beta_X(G)$ for two $(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$-discrete dynamical systems, then their state sets and update, readout functions are the same to begin with.

The machine $\beta_X(F)$ is inertial (37), because we can factor its output sheaf map p^{o} through $\mathcal{S} \to \mathsf{Ext}_1 \mathcal{X}^{\mathrm{out}}$, sending $(x, s) \in \mathcal{S}(n)$ to the sequence

$$\langle f^{\mathrm{rdt}} s_0, \dots, f^{\mathrm{rdt}} s_n, f^{\mathrm{rdt}}(f^{\mathrm{upd}}(x_n, s_n)) \rangle \in \mathcal{X}^{\mathrm{out}}(n + 1).$$

Since $s_{n+1} = f^{\mathrm{upd}}(x_n, s_n)$, this factorization is compatible with restrictions. This is due to the fact that every current state and input decide the subsequent output.

The machine is also total and deterministic, because the square below as in (44) is already a pullback:

$$
\begin{array}{ccc}
\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{in}}} \times S & \xrightarrow{(\pi_1, \pi_3)} & \widehat{X^{\mathrm{in}}} \times S \\
{\scriptstyle (\pi_1, \pi_2)} \downarrow & \lrcorner & \downarrow {\scriptstyle \pi_1} \\
\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{in}}} & \xrightarrow[\pi_1]{} & \widehat{X^{\mathrm{in}}}
\end{array}
$$

This is the case since, for each current state and input elements, the next input produces a uniquely determined state of the dynamical system, essentially because f^{upd} is a function.

Having described β_F's mapping on objects, for any map of $(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$-discrete dynamical systems $F_1 \to F_2$ described in Definition 2.3.1, the function $h \colon S_1 \to S_2$ induces a machine morphism given by $(1 \times h)^{n+1} \colon S_1(n) \to S_2(n)$, so β_X is a (faithful) functor. These are natural in X: if ϕ is a wiring diagram from X to Y, then the following commutes (up to iso, see Remark 2.1.1)

(49)
$$
\begin{array}{ccc}
\mathsf{DDS}(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}}) & \xrightarrow{\mathsf{DDS}(\phi)} & \mathsf{DDS}(\widehat{Y^{\mathrm{in}}}, \widehat{Y^{\mathrm{out}}}) \\
{\scriptstyle \beta_X} \downarrow & & \downarrow {\scriptstyle \beta_Y} \\
\mathsf{Mch}_N^{\mathrm{td}}(\mathcal{X}^{\mathrm{in}}, \mathcal{X}^{\mathrm{out}}) & \xrightarrow[\mathsf{Mch}_N^{\mathrm{td}}(K\phi)]{} & \mathsf{Mch}_N^{\mathrm{td}}(\mathcal{Y}^{\mathrm{in}}, \mathcal{Y}^{\mathrm{out}})
\end{array}
$$

Indeed, given a wiring diagram $\phi \colon X \to Y$ as in (14), the formulas (15) for $\mathsf{DDS}(\phi)$ can be applied to a discrete dynamical system $(S, f^{\mathrm{upd}}, f^{\mathrm{rdt}})$ to give rise to a new discrete system $(S, g^{\mathrm{upd}}, g^{\mathrm{rdt}})$. This is sent by β_Y to the machine

$$
\mathcal{T}(n) = \left\{ (y, s) \colon \{0, \dots, n\} \to \widehat{Y^{\mathrm{in}}} \times S \mid s_{i+1} = f^{\mathrm{upd}}(\widehat{\phi^{\mathrm{in}}}(y_i, f^{\mathrm{rdt}}(s_i)), s_i) \right\}
$$

and appropriate input and output maps. On the other hand, one can apply the formula $\mathsf{Mch}_N(\phi)$ from (20) to the machine $S \to \mathcal{X}^{\mathrm{in}} \times \mathcal{X}^{\mathrm{out}}$ from (48), i.e. take the pullback along $(\widehat{\phi^{\mathrm{in}}}, \pi_2)$ and post-compose with $1 \times \widehat{\phi^{\mathrm{out}}}$. The resulting machines $\mathcal{T} \to \mathcal{Y}^{\mathrm{in}} \times \mathcal{Y}^{\mathrm{out}}$ are isomorphic.

Finally, we can verify that $\beta_{X+Y} \circ \mathsf{DDS}_{X,Y} = (\mathsf{Mch}_N^{\mathrm{td}})_{\mathcal{X}, \mathcal{Y}} \circ (\beta_X \times \beta_Y)$ so that the natural transformation β is monoidal; this uses DDS-monoidality from Section 2.3 and (19). □

Next, we describe an analogous map that transforms continuous dynamical systems (Definition 2.3.3) to continuous machines (Definition 4.1.1). Because we will be comparing different sorts of machines, we will denote continuous machines by Mch_C rather than simply by Mch.

Let $C^{\infty} \colon \mathbf{Euc} \to \widetilde{\mathbf{Int}}$ be the functor mapping a space A to the sheaf $C^{\infty}(A)$ of C^{∞}-trajectories in X, as described in Example 3.2.6(4), which has as length-ℓ sections all smooth functions $f \colon [0, \ell] \to A$. This functor preserves all finite limits, and it induces a strong monoidal functor $\mathcal{W}_{C^{\infty}} \colon \mathcal{W}_{\mathbf{Euc}} \to \mathcal{W}_{\widetilde{\mathbf{Int}}}$ again by (9).

Proposition 5.1.2. *There exists a wiring diagram algebra map*

Proof. We first define the components $\delta_X \colon \mathsf{CDS}(X) \to \mathsf{Mch}_C(\mathcal{W}_{C^\infty}(X))$ for each box $X = (X^{\mathrm{in}}, X^{\mathrm{out}}) \in \mathcal{W}_{\mathbf{Euc}}$, mapping a continuous dynamical system on X to a continuous machine on $\mathcal{W}_{C^\infty}(X)$. We then check that the δ_X satisfy the necessary naturality and monoidality conditions. Denote by $\mathcal{X}^{\mathrm{in}}$, $\mathcal{X}^{\mathrm{out}}$ the sheaves $C^\infty(\widehat{X^{\mathrm{in}}}), C^\infty(\widehat{X^{\mathrm{out}}})$ associated to the box $\mathcal{W}_{C^\infty}(X)$ from Example 3.2.6(4).

By Definition 2.3.3 a continuous dynamical system on X is a tuple $F = (S, f^{\mathrm{dyn}}, f^{\mathrm{rdt}})$, where S is a smooth manifold, $f^{\mathrm{dyn}} \colon \widehat{X^{\mathrm{in}}} \times S \to TS$ is the dynamics, and $f^{\mathrm{rdt}} \colon S \to \widehat{X^{\mathrm{out}}}$ is a smooth map. Define a sheaf $\mathcal{S} \in \widetilde{\mathbf{Int}}$ by

$$\mathcal{S}(\ell) \coloneqq \left\{ (x, s) \colon [0, \ell] \to \widehat{X^{\mathrm{in}}} \times S \,\middle|\, x, s \text{ are smooth and } \frac{ds}{dt} = f^{\mathrm{dyn}}(x, s) \right\}$$

where $\frac{ds}{dt}$ is the derivative of $s \colon [0, \ell] \to S$; notice that the trajectory s is a solution to the differential equation defining the dynamical system. We can now define $\delta_X(F)$ to be the span $\mathcal{X}^{\mathrm{in}} \leftarrow \mathcal{S} \to \mathcal{X}^{\mathrm{out}}$ where the maps send (x, s) to x and to $f^{\mathrm{rdt}}(s)$ respectively. Again, if $\delta_X(F) = \delta_X(G)$ for two continuous dynamical systems, from the above description we can deduce that $F = G$.

We next define δ_X on morphisms, using the chain rule: consider a map of $(\widehat{X^{\mathrm{in}}}, \widehat{X^{\mathrm{out}}})$-continuous dynamical systems $F_1 \to F_2$ as defined in Definition 2.3.3, namely a smooth function $h \colon S_1 \to S_2$ such that $f_1^{\mathrm{rdt}} \circ h = f_2^{\mathrm{rdt}}$ and the right-hand square commutes in

$$
\begin{array}{ccccc}
[0, \ell] & \xrightarrow{(x, s)} & \widehat{X^{\mathrm{in}}} \times S_1 & \xrightarrow{1 \times h} & \widehat{X^{\mathrm{in}}} \times S_2 \\
& \searrow{\scriptstyle \frac{ds}{dt}} & \downarrow{\scriptstyle f_1^{\mathrm{dyn}}} & & \downarrow{\scriptstyle f_2^{\mathrm{dyn}}} \\
& & TS_1 & \xrightarrow[\frac{dh}{ds}]{} & TS_2
\end{array}
$$

We define a machine morphism $\delta_X(F_1) \to \delta_X(F_2)$, i.e. a sheaf map $\mathcal{S}_1 \to \mathcal{S}_2$ over $\mathcal{X}^{\mathrm{in}} \times \mathcal{X}^{\mathrm{out}}$ as follows. For any $(x, s) \in \mathcal{S}_1(\ell)$, the left-hand triangle commutes, i.e. $\frac{ds}{dt} = f_1^{\mathrm{dyn}}(x, s)$. Thus the outer shape does too, so we find that $(x, h(s)) \in \mathcal{S}_2(\ell)$ by the chain rule:

$$f_2^{\mathrm{dyn}}\big(x, h(s)\big) = \frac{dh}{ds} \frac{ds}{dt} = \frac{d(h \circ s)}{dt}.$$

This map commutes with the projections to $\mathcal{X}^{\mathrm{in}} \times \mathcal{X}^{\mathrm{out}}$, therefore δ_X is a (faithful) functor.

We will now show naturality and monoidality of δ, similarly to (49). Suppose $\phi \colon X \to Y$ like (14). The new machine $T \coloneqq \mathsf{Mch}_C(\phi)(\delta_X(F))$ is defined by

$$\mathcal{T}(\ell) = \left\{ (y, s) \colon [0, \ell] \to \widehat{Y^{\mathrm{in}}} \times S \,\middle|\, \frac{ds}{dt} = f^{\mathrm{dyn}}\big(s, \widehat{\phi^{\mathrm{in}}}(y, f^{\mathrm{rdt}}(s))\big) \right\}$$

by applying construction (20): first take $\mathcal{S}$'s pullback along $(\widehat{\phi^{\mathrm{in}}}, \pi_2)$ and the post-compose with $(1 \times \widehat{\phi^{\mathrm{out}}})$. This composite $\mathcal{T} \to \mathcal{Y}^{\mathrm{in}} \times \mathcal{Y}^{\mathrm{out}}$ is equal to $\delta_X(\mathsf{CDS}(\phi)(F))$ as can be seen

through (16). It can also be verified that δ is a monoidal transformation, by checking that $\delta_{X+Y} \circ \mathsf{CDS}_{X,Y} = \mathsf{Mch}_{\mathfrak{X},\mathfrak{Y}} \circ (\delta_X \times \delta_Y)$. $\square$

Remark 5.1.3. Unlike Proposition 5.1.1, the algebra map δ from Proposition 5.1.2 does not factor through $\mathsf{Mch}_C^{\mathsf{td}}$, meaning that continuous dynamical systems do not generally correspond to total and deterministic continuous machines. For example, the dynamical system $\dot{y} = y^2$ (inhabiting the closed box $\square$ with no inputs and no outputs) is not total. Indeed, the initial value $y(0) = 1$ extends to a solution $y = \frac{1}{1-t}$ that exists only for $t < 1$. Similarly, the dynamical system $\dot{y} = 2\sqrt{|y|}$ is not deterministic. Indeed, the initial value $y(0) = 0$ has the following solution for any a:

$$\begin{cases} y = 0 & \text{if } t \leq a \\ y = (t-a)^2 & \text{if } t \geq a \end{cases}$$

5.2. **Maps between machines.** We begin by providing maps between machines over the toposes of continuous, discrete, and synchronous sheaves discussed in Chapter 4. These algebra maps naturally group together according to whether they refer to general machines, or to total or deterministic variations; in fact they are morphisms of $\mathcal{WD}\text{-}\mathbf{Alg}$ in the more general setting described in Section 2.4.

As mentioned earlier, for $\mathcal{C} = \widetilde{\mathbf{Int}}$, $\widetilde{\mathbf{Int}}_N$, and $\widetilde{\mathbf{Int}}_{\mathsf{Sync}}$, the construction $\mathsf{Spn}_\mathcal{C}$ of Proposition 2.4.1 gives the wiring diagram algebras Mch_C, Mch_N, and $\mathsf{Mch}_{\mathsf{Sync}}$ described at Propositions 4.1.3, 4.4.3 and 4.4.5. Moreover, since the functors $\Sigma_i' \colon \widetilde{\mathbf{Int}}_N \to \widetilde{\mathbf{Int}}/\mathsf{Sync}$ from (31) and $\Delta_{\mathsf{Sync}!} \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}/\mathsf{Sync}$ from (32) between the respective typing categories preserve finite limits, they directly induce monoidal natural transformations $\mathsf{Spn}_{\Sigma_i'}$ and $\mathsf{Spn}_{\Delta_{\mathsf{Sync}!}}$ by Proposition 2.4.4.

Corollary 5.2.1. *There exist algebra maps realizing any continuous or discrete machine as a synchronous one*:
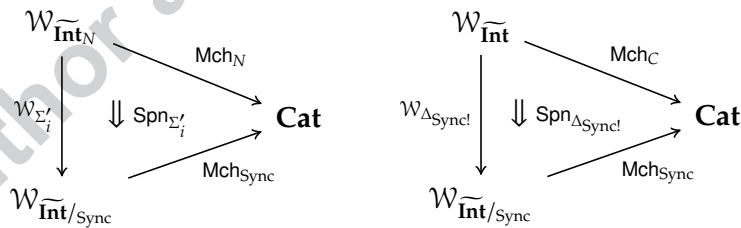
Essentially, the first algebra morphism maps a discrete machine $p \colon S \to \widehat{X^{\mathsf{in}}} \times \widehat{X^{\mathsf{out}}}$ to $\Sigma_i' p$, a synchronous machine whose state, input and output sheaves are obtained from the old ones by applying $\Sigma_i((-) \xrightarrow{!} \mathbf{1})$, see Proposition 3.3.4. Similarly, the second one gives the mapped span under $\Delta_{\mathsf{Sync}!}$, where $\Delta_{\mathsf{Sync}!}(X) = (X \times \mathsf{Sync} \to \mathsf{Sync})$ for the synchronizing sheaf (25). Notice that since Σ_i' and $\Delta_{\mathsf{Sync}!}$ are both faithful functors (see proof of Proposition 3.3.7), their respective algebra maps have embeddings as components as explained after Proposition 2.4.4.

This significant result accomplishes one of the main goals of this work, by bringing discrete and continuous systems in a common environment, that of synchronous systems. For example, both discrete and continuous dynamical systems, which can be realized as discrete and continuous machines respectively by Propositions 5.1.1 and 5.1.2, can now be

further translated under the above algebra maps to synchronous machines. As a result, their arbitrary interconnections can be studied in this common framework.

Remark 5.2.2. Using Proposition 2.4.4 we can also recover the fact from [Spi15] that extracting the steady states of a dynamical system, and organizing them in terms of matrices, amounts to an algebra homomorphism. Indeed, this follows from the obvious fact that for any finitely complete category $\mathcal{C}$ and object $c \in \mathcal{C}$, the Hom-functor $\mathcal{C}(c, -) \colon \mathcal{C} \to \mathbf{Set}$ is finitely complete, so we get a map of wiring diagram algebras

$$\mathsf{Spn}_{\mathcal{C}(c,-)} \colon \mathsf{Spn}_{\mathcal{C}} \to \mathsf{Spn}_{\mathbf{Set}}.$$

For example if $\mathcal{C} = \widetilde{\mathbf{Int}}$ and $c = \{*\}$ is the terminal sheaf (constant on one generator) then $\mathcal{C}(\{*\}, -)$ extracts the set of constant sections from any sheaf, and $\mathsf{Spn}_{\mathcal{C}(\{*\}, -)}$ is the steady state extraction from [Spi15]. If instead $c = \mathsf{Sync}$ is the synchronizing sheaf, we extract periodic cycles of length 1.

Turning to maps preserving totality and determinism, recall that total and deterministic machines require a notion of extension: for any extension of input, there exists (or is at most one) an extension of state to match. We now abstract all the necessary structure to make sense of this at a higher level of generality. Doing so will allow us to express conditions of inertiality and totality/determinism from Proposition 4.2.1 and Definition 4.2.4, for general span algebras like Proposition 2.4.1. We begin with a preliminary definition.

Definition 5.2.3. We define a *Euclidean poset* to be a poset $(E, \leq)$ equipped with a symmetric monoidal structure $(+, 0)$ such that

- the unit is minimal: $0 \leq e$ for all $e \in E$
- for all $a, b \in E$, if $b \neq 0$ there exists $N \in \mathbb{N}$ and $r \in E$ such that

$$a = N \cdot b + r \quad \text{and} \quad r \leq b$$

where $N \cdot b := b + \cdots + b$ (N times).

A *morphism* of Euclidean posets is a strong monoidal functor.

Lemma 5.2.4. *Suppose $v \colon E \to E'$ is a morphism of Euclidean posets. If there exists $b \neq 0$ such that $v(b) = 0$ then $v(a) = 0$ for all a; such a morphism is called* trivial, *otherwise it is* nontrivial.

To cast totality at this level of generality, we need our categories and functors to be regular, rather than just finitely-complete; i.e. we restrict our general $\mathcal{C}$-span system functor $\mathsf{Spn}_{(-)} \colon \mathbf{FCCat} \to \mathcal{WD}\text{-}\mathbf{Alg}$ (22) to $\mathbf{RegCat}$, the 2-category of regular categories and functors, with all natural transformations between them. Recall that every topos is a regular category and every epimorphism in a topos is regular. See [Bar71] or [Joh02] for more on regular categories and functors.

For any regular category $\mathcal{C}$, $(\mathsf{End}(\mathcal{C}), \circ, 1_{\mathcal{C}})$ is the monoidal category of regular endofunctors on $\mathcal{C}$. The following generalize the respective constructions from Section 4.2.

Definition 5.2.5. For $\mathcal{C} \in \mathbf{RegCat}$, an *extension structure* on $\mathcal{C}$ is a pair (E, Ext), where $(E, \leq, +, 0)$ is a Euclidean poset and $\mathsf{Ext} \colon E^{\mathrm{op}} \to \mathsf{End}(\mathcal{C})$ is a strong monoidal functor.

Spelling out what it means for $\mathsf{Ext} \colon E^{\mathrm{op}} \to \mathsf{End}(\mathcal{C})$ to be strong monoidal,

- for all $e \in E$, the functor $\mathsf{Ext}_e \colon \mathcal{C} \to \mathcal{C}$ preserves finite limits and regular epimorphisms;
- $\mathsf{Ext}_0 = \mathrm{id}_{\mathcal{C}}$, and for all $a, b \in E$, we have $\mathsf{Ext}_{a+b} = \mathsf{Ext}_a \circ \mathsf{Ext}_b$;
- for any $e' \le e$ there is a chosen map which we will denote $\lambda_{e,e'} \colon \mathsf{Ext}_e \Rightarrow \mathsf{Ext}_{e'}$. When $e' = 0$ we abbreviate $\lambda_{e,0}$ by $\lambda_e \colon \mathsf{Ext}_e \Rightarrow \mathrm{id}_{\mathcal{C}}$; and
- for all $a, b \in E$, there exist $N \in \mathbb{N}$ and $r \in E$ such that $\mathsf{Ext}_a = \mathsf{Ext}_{N \cdot b} \circ \mathsf{Ext}_r$.

Using these extension structures we can formalize inertiality for the $\mathcal{C}$-span systems of Definition 2.4.2, generalizing Definition 4.2.4. Notice that we denote the components $(\lambda_e)_A \colon \mathsf{Ext}_e A \to A$ of the natural transformation λ_e by the same name, in order to avoid double subscripts; the (co)domain should clarify each component.

Definition 5.2.6. Given an extension structure on $\mathcal{C}$, we say that a map $p \colon S \to A$ in $\mathcal{C}$ is Ext-*inertial* if there exists $0 \ne e \in E$ such that p factors through $\lambda_e \colon \mathsf{Ext}_e(A) \to \mathsf{Ext}_0(A) = A$. For any box $X \in \mathcal{W}_{\mathcal{C}}$, we say that a $\mathcal{C}$-span $(p^{\mathrm{i}}, p^{\mathrm{o}}) \colon S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$ is Ext-*inertial* if p^{o} is. Denote $\mathsf{Spn}_{\mathcal{C}}^{\mathrm{in}}(X) \subseteq \mathsf{Spn}_{\mathcal{C}}(X)$ the full subcategory of all Ext-inertial $\mathcal{C}$-span systems on X.

We can also generalize totality and determinism, using the following lemma (compare to Proposition 4.2.1).

Lemma 5.2.7. *Let $\mathcal{C}$ be a regular category with extension structure $\mathsf{Ext} \colon E^{\mathrm{op}} \to \mathsf{End}(\mathcal{C})$. Given $e \in E$ and $p \colon S \to A$, define $h_e \colon \mathsf{Ext}_e S \to S_e$ to be the universal map to the pullback denoted S_e in*

(50)

$$
\begin{array}{ccc}
\mathsf{Ext}_e S \xrightarrow{\quad \lambda_e \quad} & & \\
\downarrow{\scriptstyle h_e} & & \\
S_e \xrightarrow{\ \lambda'_e\ } S & \\
\downarrow{\scriptstyle p_e} \quad \lrcorner \quad \downarrow{\scriptstyle p} & \\
\mathsf{Ext}_e A \xrightarrow[\ \lambda_e\ ]{} A &
\end{array}
$$

Then the following are equivalent:

(1) *for all $e \in E$, the map $h_e \colon \mathsf{Ext}_e S \to S_e$ is a regular epimorphism (resp. a monomorphism, an isomorphism);*

(2) *there exists $0 \ne d \in E$ such that for all $d' \le d$, the map $h_{d'} \colon \mathsf{Ext}_{d'} S \to S_e$ is a regular epimorphism (resp. a monomorphism, an isomorphism).*

Proof. Clearly 1⇒2, so assume the latter holds for $d \ne 0$. By the assumption that E is a Euclidean poset, it suffices to show that for any $a, b \in E$ and $p \colon S \to A$, if both h_a and h_b have property **P**, where **P** is the property of being a regular epimorphism (resp. a monomorphism, an isomorphism) then so does h_{a+b}.

Consider the two diagrams below, where the left-hand diagram is as in (50):

$$
\begin{array}{c}
\text{Ext}_{a+b}S \\
\downarrow{\scriptstyle \text{Ext}_b h_a} \\
\text{Ext}_b S_a \xrightarrow{\text{Ext}_b \lambda_a} \text{Ext}_b S
\end{array}
$$

$$
\begin{array}{ccc}
\text{Ext}_a S & & \\
\downarrow{\scriptstyle h_a} & & \\
S_a \xrightarrow{\lambda'_a} S & & \\
\downarrow{\scriptstyle p_a}\ \lrcorner\ \ \downarrow{\scriptstyle p} & & \\
\text{Ext}_a A \xrightarrow{\lambda_a} A & &
\end{array}
$$
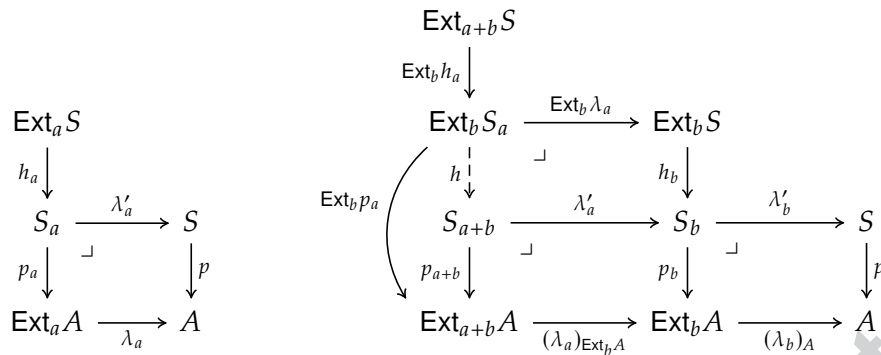
For the right-hand diagram, begin by forming the two bottom pullback squares. By definition of h_b (50), we have $\text{Ext}_b p = p_b \circ h_b$, so the vertical rectangle is an application of Ext_b to the left-hand diagram, and is thus a pullback because Ext_b is regular. There is an induced map $h\colon \text{Ext}_b S_a \to S_{a+b}$, and we now have that each indicated square in the diagram is a pullback.

Since $\text{Ext}_{a+b}p = \text{Ext}_b\text{Ext}_a p = \text{Ext}_b p_a \circ \text{Ext}_b h_a$ is the long vertical map, the universal property of pullback implies that $h \circ \text{Ext}_b h_a = h_{a+b}$. If h_a and h_b have property **P** then so does h_{a+b} because its factors h and $\text{Ext}_b h_a$ do: **P** is stable under pullbacks and Ext_b preserves **P**. $\qquad\square$

Definition 5.2.8. Let $\mathcal{C}$ be a regular category with extension structure $\text{Ext}\colon E^{\text{op}} \to \text{End}(\mathcal{C})$, and let $p\colon S \to B$ be a morphism. We say that p is Ext-*total* (resp. Ext-*deterministic*, Ext-*total-determinstic*) if the map h_e in (50) is a regular epimorphism (resp. a monomorphism, an isomorphism) for all $e \in E$.

For any box $X \in \mathcal{W}_\mathcal{C}$, we call an Ext-inertial span system $(p^{\text{i}}, p^{\text{o}}) \in \text{Spn}_\mathcal{C}^{\text{in}}(X)$ (as in Definition 5.2.6) Ext-*total* (resp. Ext-*deterministic*, Ext-*total-deterministic*) if p^{i} is. Denote $\text{Spn}_\mathcal{C}^{\text{t}}(X), \text{Spn}_\mathcal{C}^{\text{d}}(X), \text{Spn}_\mathcal{C}^{\text{td}}(X) \subseteq \text{Spn}_\mathcal{C}^{\text{in}}(X)$ the full subcategories of all Ext-total (resp. Ext-deterministic, Ext-total-deterministic) span systems on X.

Example 5.2.9. For any $\epsilon \geq 0$ the functor $\text{Ext}_\epsilon\colon \widetilde{\textbf{Int}} \to \widetilde{\textbf{Int}}$, as in Definition 3.2.8, is an endomorphism of $\widetilde{\textbf{Int}}$; we showed it is finitely complete in Lemma 3.2.9, and it is not hard to check that it preserves epimorphisms (all of which are regular). The poset $E = (\mathbb{R}_{>0}, \geq)$ of positive reals is a Euclidean poset, and the left restriction maps $\text{Ext}_\epsilon \to \text{Ext}_{\epsilon'}$ for any $\epsilon \geq \epsilon'$ constitute an extension structure $\text{Ext}\colon E^{\text{op}} \to \text{End}(\widetilde{\textbf{Int}})$. This extension structure can be lifted to one on $\widetilde{\textbf{Int}}/\text{Sync}$. Namely, for every $e \in \mathbb{R}_{\geq 0}$, we define $\text{Ext}_e(X \to \text{Sync})$ to be the composite $\text{Ext}_e X \to \text{Ext}_e(\text{Sync}) \xrightarrow{\lambda_e} \text{Sync}$. There is also an extension structure on $\widetilde{\textbf{Int}}_N$, such that Definition 5.2.5 generalizes the definitions in Section 4.4, by taking $E = \mathbb{N}$.

In all the above cases, the notions of inertial, total, deterministic, and total-deterministic morphisms and span systems generalize the older notions, as aimed.

Proposition 5.2.10. *Let $\mathcal{C}$ be a regular category with an extension structure* $\text{Ext}\colon E^{\text{op}} \to \text{End}(\mathcal{C})$. *Then there are symmetric lax monoidal functors*

$$
\text{Spn}_\mathcal{C}^{\text{in}},\ \text{Spn}_\mathcal{C}^{\text{t}},\ \text{Spn}_\mathcal{C}^{\text{d}},\ \text{Spn}_\mathcal{C}^{\text{td}}\colon \mathcal{W}_\mathcal{C} \longrightarrow \textbf{Cat}
$$

defined as in Definitions 5.2.6 and 5.2.8 on any $X \in \mathcal{W}_\mathcal{C}$, which are subalgebras of $\text{Spn}_\mathcal{C}$ (17).

Proof. The proof for inertiality closely follows that of Proposition 4.3.1, and totality and determinism follow that of Proposition 4.3.2.   □

These subfunctors constitute, on their own right, the mapping of a more general functor on objects, namely regular categories. Towards that end, we define the following category.

Definition 5.2.11. Let $(\mathcal{C}, E, \mathsf{Ext})$ and $(\mathcal{C}', E', \mathsf{Ext}')$ be regular categories with extension structures. A *morphism* between them is a pair (F, ν) where $F \colon \mathcal{C} \to \mathcal{C}'$ is a regular functor, $\nu \colon E \to E'$ is a nontrivial morphism of Euclidean posets, and such that for all $e \in E$ the following diagram commutes (up to natural isomorphism)

$$
\begin{array}{ccc}
\mathcal{C} & \xrightarrow{\ \mathsf{Ext}_e\ } & \mathcal{C} \\
{\scriptstyle F}\downarrow & & \downarrow{\scriptstyle F} \\
\mathcal{C}' & \xrightarrow[\mathsf{Ext}'_{\nu(e)}]{} & \mathcal{C}'
\end{array}
$$

These form the category of *regular categories with extensions*, denoted $\mathbf{RegCat}_e$, which naturally maps to $\mathbf{RegCat}_e \to \mathbf{RegCat} \subseteq \mathbf{FCCat}$.

Theorem 5.2.12. *The restricted functor* $\mathsf{Spn}_{(-)} \colon \mathbf{RegCat}_e \to \mathbf{RegCat} \hookrightarrow \mathbf{FCCat} \to \mathcal{WD}\text{-}\mathbf{Alg}$ *as in Proposition 2.4.4 has subfunctors*

$$
\mathsf{Spn}^{\mathrm{in}}_{(-)}, \mathsf{Spn}^{\mathrm{t}}_{(-)}, \mathsf{Spn}^{\mathrm{d}}_{(-)}, \mathsf{Spn}^{\mathrm{td}}_{(-)} \colon \mathbf{RegCat}_e \to \mathcal{WD}\text{-}\mathbf{Alg}
$$

which map any object $\mathsf{Ext}\colon E^{\mathrm{op}} \to \mathsf{End}(\mathcal{C})$ *in* $\mathbf{RegCat}_e$ *to the respective algebras of Proposition 5.2.10.*

Proof. For their mapping on morphisms, given some $(F, \nu)\colon (\mathcal{C}, E, \mathsf{Ext}) \to (\mathcal{C}', E', \mathsf{Ext}')$ as in Definition 5.2.11, we need to show that Spn_F of (23) appropriately restricts, through its component functors, to a 2-cell of the form

$$
\begin{array}{ccc}
\mathcal{W}_{\mathcal{C}} & & \\
{\scriptstyle \mathcal{W}_F}\downarrow \quad \Downarrow {\scriptstyle \mathsf{Spn}^{\mathsf{x}}_F} & \searrow^{\mathsf{Spn}^{\mathsf{x}}_{\mathcal{C}}} & \\
& & \mathbf{Cat} \\
\mathcal{W}_{\mathcal{C}'} & \nearrow_{\mathsf{Spn}^{\mathsf{x}}_{\mathcal{C}'}} &
\end{array}
$$

where $\mathsf{x} \in \{\mathsf{i}, \mathsf{t}, \mathsf{d}, \mathsf{td}\}$ stands for are all respective subalgebras of $\mathsf{Spn}_{\mathcal{C}}$. Therefore it suffices to show that in all three cases, a dashed arrow exists for any $X \in \mathcal{W}_{\mathcal{C}}$, making it commute in **Cat**:

(51)
$$
\begin{array}{ccc}
\mathsf{Spn}^{\mathsf{x}}_{\mathcal{C}}(X) & \dashrightarrow & \mathsf{Spn}^{\mathsf{x}}_{\mathcal{C}'}(\mathcal{W}_F X) \\
\downarrow & & \downarrow \\
\mathsf{Spn}_{\mathcal{C}}(X) & \xrightarrow[(\mathsf{Spn}_F)_X]{} & \mathsf{Spn}_{\mathcal{C}'}(\mathcal{W}_F X)
\end{array}
$$

For inertial machines, the existence of a dashed arrow is implied by the fact that ν is nontrivial and $F\mathsf{Ext}_e \cong \mathsf{Ext}_{\nu(e)}F$ for all $e \in E$. For total (resp. deterministic) machines, the existence of a dashed arrow is implied by the fact that Ext_e preserves regular epimorphisms (resp. monomorphisms) for all e. Moreover, notice that if $(\mathsf{Spn}_F)_X$ is an embedding, then clearly so is $(\mathsf{Spn}^{\mathsf{x}}_F)_X$.   □

We can now apply Theorem 5.2.12 to the regular categories $\widetilde{\mathbf{Int}}, \widetilde{\mathbf{Int}}_N$ and $\widetilde{\mathbf{Int}}/\mathrm{Sync}$ from Sections 3.2 and 3.3, and to the functors $\Sigma'_i \colon \widetilde{\mathbf{Int}}_N \to \widetilde{\mathbf{Int}}/\mathrm{Sync}$ (31) and $\Delta_{\mathrm{Sync!}} \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}/\mathrm{Sync}$ (32), in order to obtain algebra maps between the total and deterministic variations of continuous, discrete and synchronous machines, as anticipated. Recall that as defined in Chapter 4, the category of e.g. continuous machines is precisely the category of $\widetilde{\mathbf{Int}}$-span systems $\mathsf{Spn}_{\widetilde{\mathbf{Int}}}$, denoted Mch_C; similarly for the rest of terminology.

Corollary 5.2.13. *Let* $\mathbf{x} \in \{\mathsf{i}, \mathsf{t}, \mathsf{d}, \mathsf{td}\}$ *stand for inertial, total, deterministic, or total-deterministic. There are algebra embeddings*

which translate the specific classes of discrete machines into synchronous machines of the same class, and similarly for continuous into synchronous.

Proof. First of all, notice that both $\Sigma'_i \colon \widetilde{\mathbf{Int}}_N \to \widetilde{\mathbf{Int}}/\mathrm{Sync}$ and $\Delta_{\mathrm{Sync!}} \colon \widetilde{\mathbf{Int}} \to \widetilde{\mathbf{Int}}/\mathrm{Sync}$ are inverse image parts of geometric morphisms, so they are regular functors. Clearly $(\Delta_{\mathrm{Sync!}}, \mathrm{id}_{\mathbb{R}_{\geq 0}})$ is a map in $\mathbf{RegCat}_e$, so the second algebra maps follow directly from Theorem 5.2.12.

Now the inclusion $\nu \colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is a nontrivial morphism of Euclidean posets, so to complete the proof, we need to show that $\Sigma'_i \circ \mathrm{Ext}_n \cong \mathrm{Ext}_{\nu(n)} \circ \Sigma'_i$ for any $n \in \mathbb{N}$. It suffices to show this for $n = 1$, whence $\nu(n) = 1$. Recall from Proposition 3.3.4 the formula

$$\Sigma_i X(\ell) \cong \bigsqcup_{r \in [0,1)} X(\lceil r + \ell \rceil)$$

where the extra data of $\Sigma'_i X$ is just the map $\Sigma_X \to \mathrm{Sync}$ given by r. Thus the result follows from the equation $\lceil \ell + r + 1 \rceil = \lceil \ell + r \rceil + 1$. □

Notice that by construction of the above algebra morphisms, their components commute with the embeddings of each subclass of machines into the general discrete or continuous ones, as in (51).

Remark 5.2.14. It should be noted that the only reason we need to work with regular categories and functors rather than finitely complete ones, is for totalness. We can replace $\mathbf{RegCat}$ by $\mathbf{FCCat}$ and Theorem 5.2.12 will still hold for inertial, deterministic, and total-deterministic machines. That is, in Definition 5.2.5, we could ask only that $\mathcal{C}$ be finitely complete, and that endofunctors $(\mathcal{C} \to \mathcal{C}) \in \mathrm{End}(\mathcal{C})$ preserve finite limits. Similarly we can drop condition that our finitely-complete functors $F \colon \mathcal{C} \to \mathcal{C}'$ are regular; going through with appropriate changes to all constructions above would only exclude totalness results.

The above Corollary 5.2.13 successfully restricts the more general Corollary 5.2.1 to the total and deterministic variations of continuous, discrete and synchronous machines. Once again, (total, deterministic) synchronous machines end up being the common framework

where their discrete and continuous counterparts can be studied together. The final proposition below ensures that contracted machines described in Section 4.5 fit in the same picture.

Proposition 5.2.15. *There exists a functor* $\mathsf{Cntr}_{(-)}\colon \mathbf{RegCat} \to \mathcal{W}\mathbf{D\text{-}Alg}$ *making the diagram*

$$
\begin{array}{ccc}
 & & \mathcal{W}\mathbf{D\text{-}Alg} \\
 & \nearrow{\scriptstyle\mathsf{Cntr}_{(-)}} & \downarrow{\scriptstyle U} \\
\mathbf{RegCat} & \xrightarrow[\ \mathcal{W}_{(-)}\ ]{} & \mathcal{SMC}
\end{array}
$$

commute. Moreover, there exists a natural transformation

$$
\begin{array}{ccc}
 & \mathbf{FCCat} & \\
\nearrow & \xRightarrow{\ \mathsf{Img}\ } & \searrow{\scriptstyle \mathsf{Spn}_{(-)}} \\
\mathbf{RegCat} & \xrightarrow[\ \mathsf{Cntr}_{(-)}\ ]{} & \mathcal{W}\mathbf{D\text{-}Alg}
\end{array}
$$

whose components translate each machine (therefore any total/deterministic subclass, Theorem 5.2.12) into its 'maximal' validated safety contract.

Proof. Similarly to Proposition 2.4.4, this functor maps any regular category $\mathcal{C}$ to its algebra $\mathsf{Cntr}_{\mathcal{C}}\colon \mathcal{W}_{\mathcal{C}} \to \mathbf{Cat}$ of safety contracts essentially described in Proposition 4.5.3, i.e. $\mathsf{Cntr}_{\mathcal{C}}(X) := \mathsf{Sub}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}})$ for any box $X \in \mathcal{W}_{\mathcal{C}}$. For any $\phi\colon X \to Y$ in $\mathcal{C}$, the functor $\mathsf{Cntr}_{\mathcal{C}}(\phi)$ is given by the construction (46) (seen inside an arbitrary regular category), and the symmetric lax monoidal structure follows since products of inclusions are inclusions. Now any regular functor $F\colon \mathcal{C} \to \mathcal{D}$ preserves epi-mono factorization and pullbacks, so it induces a map

$$
\begin{array}{ccc}
\mathcal{W}_{\mathcal{C}} & & \\
\downarrow{\scriptstyle\mathcal{W}_F} \quad \xRightarrow{\ \mathsf{Cntr}_F\ } & \searrow{\scriptstyle\mathsf{Cntr}_{\mathcal{C}}} & \\
 & & \mathbf{Cat} \\
\mathcal{W}_{\mathcal{D}} & \nearrow{\scriptstyle\mathsf{Cntr}_{\mathcal{D}}} &
\end{array}
$$

with components functors $(\mathsf{Cntr}_F)_X\colon \mathsf{Cntr}_{\mathcal{C}}(X) \to \mathsf{Cntr}_{\mathcal{D}}(\mathcal{W}_F X)$ for any $X \in \mathcal{W}_C$ being just application of F on the respective subobjects.

Now the natural transformation Img has components wiring diagram algebra maps $\mathsf{Img}_{\mathcal{C}}\colon \mathsf{Spn}_{\mathcal{C}} \to \mathsf{Cntr}_{\mathcal{C}}$, formed by the mappings

$$
\mathsf{Spn}_{\mathcal{C}}(X) = \mathcal{C}/(\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) \to \mathsf{Sub}_{\mathcal{C}}(\widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}) = \mathsf{Cntr}_{\mathcal{C}}(X)
$$

which take the image of each $S \to \widehat{X^{\mathrm{in}}} \times \widehat{X^{\mathrm{out}}}$; these are functorial and symmetric lax monoidal. Finally, naturality of Img

$$
\begin{array}{ccc}
\mathsf{Spn}_{\mathcal{C}} & \xrightarrow{\ \mathsf{Spn}_F\ } & \mathsf{Spn}_{\mathcal{D}} \\
\downarrow{\scriptstyle\mathsf{Img}_{\mathcal{C}}} & & \downarrow{\scriptstyle\mathsf{Img}_{\mathcal{D}}} \\
\mathsf{Cntr}_{\mathcal{C}} & \xrightarrow[\ \mathsf{Cntr}_F\ ]{} & \mathsf{Cntr}_{\mathcal{D}}
\end{array}
$$

is verified when we write the above commutativity inside **WD-Alg**, i.e. arrows as in (13):

$$\mathcal{W}_{\mathcal{C}} \xrightarrow{\text{Spn}_{\mathcal{C}}} \mathbf{Cat}$$

(diagram with $\mathcal{W}_{\mathcal{C}}$, $\mathcal{W}_{\mathcal{D}}$, $\mathcal{W}_{\mathcal{D}}$ on the left, arrows $\mathcal{W}_F$, $\Downarrow \text{Spn}_F$, $\text{Spn}_{\mathcal{D}}$, $\Downarrow \text{Img}_{\mathcal{D}}$, $\text{Cntr}_{\mathcal{D}}$ to **Cat**) $=$ (diagram with $\mathcal{W}_{\mathcal{C}}$, $\mathcal{W}_{\mathcal{C}}$, $\mathcal{W}_{\mathcal{D}}$, arrows $\text{Spn}_{\mathcal{C}}$, $\Downarrow \text{Img}_{\mathcal{C}}$, $\text{Cntr}_{\mathcal{C}}$, $\mathcal{W}_F$, $\Downarrow \text{Cntr}_F$, $\text{Cntr}_{\mathcal{D}}$ to **Cat**)

$\square$

This last algebra map has the significant effect of translating machines of any kind to a safety contract consisting of all its 'valid' behaviors, i.e. lists of all possible inputs and outputs through time; see also Definition 4.5.1. As a result, whenever we have an interconnection (11) of arbitrary systems, we could directly reason about the valid behaviors of the composite system without composing the machines first.

APPENDIX A. DISCRETE CONDUCHÉ FIBRATIONS

In this appendix, we discuss an equivalent view of interval sheaves from Section 3.2 in terms of *discrete Conduché fibrations*, elsewhere [BF00] called *unique factorization lifting functors*. This view largely follows material found in [Joh99], and allows us to think of continuous or discrete interval sheaves as categories equipped with 'length' functors, called *durations* in [Law86], into the monoids $(\mathbb{R}_{\geq 0}, 0, +)$ and $(\mathbb{N}, 0, +)$ viewed as single-object categories.

A.1. **Discrete fibrations, opfibrations, and Conduché fibrations.** For any category $\mathcal{C}$, consider the diagram of sets and functions

$$C_2 \underset{}{\overset{\circ}{\rightrightarrows}} C_1 \xrightarrow[t]{s} C_0$$

where C_0 is the set of objects, C_1 is the set of morphisms, and C_2 is the set of composable morphisms in $\mathcal{C}$. The two functions $C_1 \to C_0$ send a morphism f to its source (domain) and its target (codomain); the three functions $C_2 \to C_1$ send a composable pair (g, f) to g, $g \circ f$, and f. We have left out of our diagram the $(i + 1)$ functions $C_i \to C_{i+1}$ induced by identity morphisms. For any functor $F : \mathcal{C} \to \mathcal{D}$, there is a function $F_i : C_i \to D_i$ for each $i \in \{0, 1, 2\}$, and each is induced by the function F_0 on objects and the function F_1 on morphisms.

Definition A.1.1. For a functor $F : \mathcal{C} \to \mathcal{D}$, consider the commutative diagrams

$$
\begin{array}{ccc}
C_1 \xrightarrow{t} C_0 & C_1 \xrightarrow{s} C_0 & C_2 \xrightarrow{\circ} C_1 \\
F_1 \downarrow \quad \downarrow F_0 & F_1 \downarrow \quad \downarrow F_0 & F_2 \downarrow \quad \downarrow F_1 \\
D_1 \xrightarrow{t} D_0 & D_1 \xrightarrow{s} D_0 & D_2 \xrightarrow{\circ} D_1
\end{array}
$$

Then F is called a *discrete fibration* (resp. *a discrete opfibration, a discrete Conduché fibration*) if the first (resp. second, third) square is a pullback.

The first two conditions clearly correspond to the well-known definitions of discrete (op)fibrations. For example, the first one says that for every morphism $h\colon x \to y$ in the base category $\mathcal{D}$ and every object $d \in \mathcal{C}$ above y, there exists a unique morphism $c \to d$ which maps to h via F. The third one, on the other hand, says that given a morphism $f\colon c \to d$ in the domain category $\mathcal{C}$ such that $Ff = v \circ u$ factorizes in the base category $\mathcal{D}$, there exists a unique factorization $f = h \circ g$ with $Fh = v$ and $Fg = u$. This is also equivalent, [Law86], to the isomorphism of the factorization categories $\mathrm{Fact}(f) \cong \mathrm{Fact}(Ff)$ from Definition 3.1.5.

Lemma A.1.2. *If F is a discrete opfibration (resp. a discrete fibration), then it is a discrete Conduché fibration.*

Proof. The top, bottom, and front of the following cube are pullbacks, so the back is too:
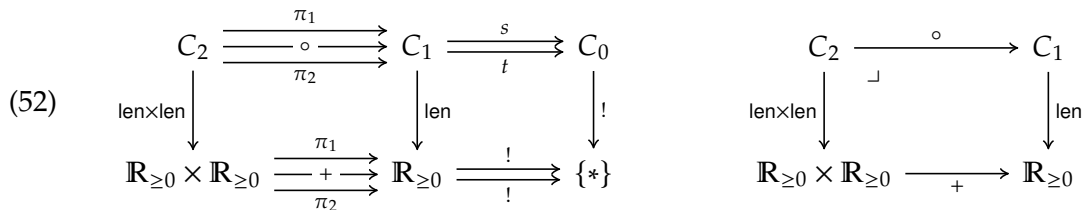
$$
\begin{array}{ccc}
C_2 & \longrightarrow & C_1 \\
& \searrow & \\
& C_1 \longrightarrow C_0 & \\
\downarrow & \downarrow & \\
& \downarrow & \\
D_2 & \longrightarrow & D_1 \\
& \searrow & \\
& D_1 \longrightarrow D_0 &
\end{array}
$$

$\square$
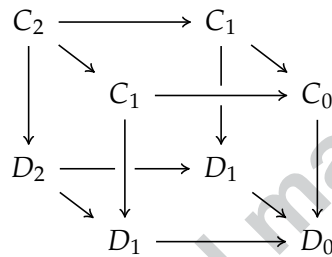
Lemma A.1.3. *If $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{E}$ are functors and G is a discrete Conduché fibration, then $G \circ F$ is discrete Conduché if and only if F is. The same holds if we replace discrete Conduché with discrete (op)fibrations.*

Proof. This is just the pasting lemma for pullback squares. $\square$

Small discrete Conduché fibrations form a wide subcategory of the category of small categories, $\mathbf{DCF} \subseteq \mathbf{Cat}$. In particular, $\mathbf{DCF}/\mathcal{A}$ denotes the slice category of discrete Conduché fibrations over any $\mathcal{A} \in \mathbf{Cat}$. Our main case of interest is the case $\mathcal{A} = \mathcal{R}$, the additive monoid of reals, which is also the primary example in the development of [Law86]. Hence a discrete Conduché fibration $\mathsf{len}\colon \mathcal{C} \to \mathcal{R}$ for any category $\mathcal{C}$ amounts to a commutative diagram as to the left

(52)

$$
\begin{array}{ccccc}
C_2 & \underset{\pi_2}{\overset{\pi_1}{\underset{\longrightarrow}{\overset{\longrightarrow}{\circ}}}} & C_1 & \overset{s}{\underset{t}{\rightrightarrows}} & C_0 \\
{\scriptstyle \mathsf{len}\times\mathsf{len}}\downarrow & & {\scriptstyle \mathsf{len}}\downarrow & & \downarrow{\scriptstyle !} \\
\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} & \underset{\pi_2}{\overset{\pi_1}{\underset{\longrightarrow}{\overset{\longrightarrow}{+}}}} & \mathbb{R}_{\geq 0} & \overset{!}{\underset{!}{\rightrightarrows}} & \{*\}
\end{array}
\qquad
\begin{array}{ccc}
C_2 & \overset{\circ}{\longrightarrow} & C_1 \\
{\scriptstyle \mathsf{len}\times\mathsf{len}}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{len}} \\
\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} & \underset{+}{\longrightarrow} & \mathbb{R}_{\geq 0}
\end{array}
$$

for which the sub-diagram extracted to its right is a pullback in **Set**. In other words, every morphism f in $\mathcal{C}$ has a length $\mathsf{len}(f) \in \mathbb{R}_{\geq 0}$ and, for any way to write $\mathsf{len}(f) = \ell_1 + \ell_2$ as a sum of nonnegative numbers, there is a unique pair of composable morphisms $f = f_1 \circ f_2$ in $\mathcal{C}$ having those lengths $\mathsf{len}(f_1) = \ell_1$ and $\mathsf{len}(f_2) = \ell_2$.

Proposition A.1.4. *For any small category* $\mathcal{C}$, *the slice category* **DCF**/$\mathcal{C}$ *is reflective in* **Cat**/$\mathcal{C}$. *Moreover, if* $F\colon \mathcal{C} \to \mathcal{D}$ *is any functor, there is a diagram*

$$
\begin{array}{ccc}
\mathbf{DCF}/\mathcal{C} & \underset{\Delta_F}{\overset{\Sigma_F}{\underset{\perp}{\rightleftarrows}}} & \mathbf{DCF}/\mathcal{D} \\[2mm]
L_{\mathcal{C}} \dashv \; U_{\mathcal{C}} \big\updownarrow & & L_{\mathcal{D}} \dashv \; U_{\mathcal{D}} \big\updownarrow \\[2mm]
\mathbf{Cat}/\mathcal{C} & \underset{F^*}{\overset{F\circ(-)}{\underset{\perp}{\rightleftarrows}}} & \mathbf{Cat}/\mathcal{D}
\end{array}
$$

which commutes for the right (*and hence left*) *adjoints, where* F^* *is given by pullback along* F.

Proof. The existence of a left adjoint $L_{\mathcal{C}}$ to the inclusion $U_{\mathcal{C}}\colon \mathbf{DCF}/\mathcal{C} \to \mathbf{Cat}/\mathcal{C}$ is proven in [Joh99, Prop. 1.3], by showing that $U_{\mathcal{C}}$ preserves all limits and satisfies a solution set condition.

If $\mathcal{A} \to \mathcal{D}$ is a discrete Conduché fibration, then its pullback $F^*(\mathcal{A}) \to \mathcal{C}$ is also discrete Conduché by the pasting lemma for pullbacks. Hence Δ_F is defined as the restriction of F^* on **DCF**/$\mathcal{D}$, and $F^* \circ U_{\mathcal{D}} = U_{\mathcal{C}} \circ \Delta_F$.

Let $F_! = F\circ(-)\colon \mathbf{Cat}/\mathcal{C} \to \mathbf{Cat}/\mathcal{D}$ be the left adjoint of F^*. We define $\Sigma_F\colon \mathbf{DCF}/\mathcal{C} \to \mathbf{DCF}/\mathcal{D}$ to be the composite $L_{\mathcal{D}} \circ F_! \circ U_{\mathcal{C}}$. Using the fact that $U_{\mathcal{C}}$ is fully faithful, a calculation shows that Σ_F is indeed left adjoint to Δ_F:

$$[\Sigma_F \mathcal{A}, \mathcal{B}] = [(L_{\mathcal{D}} \circ F_! \circ U_{\mathcal{C}})\mathcal{A}, \mathcal{B}] \cong [U_{\mathcal{C}}\mathcal{A}, (F^* \circ U_{\mathcal{D}})\mathcal{B}] \cong [U_{\mathcal{C}}\mathcal{A}, (U_{\mathcal{C}} \circ \Delta_F)\mathcal{B}] = [\mathcal{A}, \Delta_F \mathcal{B}]. \quad \square$$

A.2. **The equivalence** $\widetilde{\mathbf{Int}} \cong \mathbf{DCF}/\mathcal{R}$. Having discussed discrete Conduché fibrations and their properties, we are ready to show that the topos of interval sheaves is a special case. The notion of a factorization-linear category $\mathcal{C}$ (Definition 3.1.5) turns out to capture all the necessary structure for the slice category **DCF**/$\mathcal{C}$ to be a sheaf topos.

Theorem A.2.1. [Joh99, Prop. 3.6] *Suppose that* $\mathcal{C}$ *is a factorization-linear category, let* $\mathcal{C}_{\mathrm{tw}}$ *be its twisted arrow category with its Johnstone coverage* (*Definition* 3.2.1), *and let* $\widetilde{\mathcal{C}_{\mathrm{tw}}}$ *be the associated sheaf topos. Then there is an equivalence of categories*

$$\mathbf{DCF}/\mathcal{C} \simeq \widetilde{\mathcal{C}_{\mathrm{tw}}}.$$

Corollary A.2.2. *There is an equivalence between the topos of continuous sheaves* (*resp. discrete-interval sheaves*) *and discrete Conduché fibrations over* $\mathcal{R}$ (*resp. over* $\mathcal{N}$):

(53) $$\mathbf{DCF}/\mathcal{R} \simeq \widetilde{\mathbf{Int}} \quad \text{and} \quad \mathbf{DCF}/\mathcal{N} \simeq \widetilde{\mathbf{Int}}_N$$

Remark A.2.3. Note that Proposition A.1.4 does not follow from Proposition 3.3.3, even though $\mathbf{DCF}/\mathcal{N} \simeq \widetilde{\mathbf{Int}}_N$ and $\mathbf{DCF}/\mathcal{R} \simeq \widetilde{\mathbf{Int}}$ and the upper adjunctions are essentially the same; this is because discrete Conduché fibrations and sheaves give different perspectives. Categories emphasize composition, and adding the Conduché condition enforces that morphisms can be factorized. Conversely, presheaves emphasize restriction, and adding the sheaf condition enforces that sections can be glued.

These two perspectives compare as follows. Let **wGrph** denote the category of weighted graphs (with nonnegative edge weights), i.e. objects are $G = \{E \rightrightarrows V, E \to \mathbb{R}_{\geq 0}\}$. If we define $\mathcal{I}$ to be the category with objects $\mathbb{R}_{\geq 0} \sqcup \{v\}$ and two morphisms $s_\ell, t_\ell\colon v \to \ell$ for each $\ell \in \mathbb{R}_{\geq 0}$, then we have **wGrph** $\cong \mathsf{Psh}(\mathcal{I})$.

There is a functor $\mathcal{I} \to \mathbf{Int}$ sending $v \mapsto 0$ and $\ell \mapsto \ell$ for all $\ell \in \mathbb{R}_{\geq 0}$, and sending $s_\ell \mapsto \mathsf{Tr}_0$ and $t_\ell \mapsto \mathsf{Tr}_\ell$, the left and right endpoints. This induces a left Kan extension between the presheaf categories, $\mathbf{wGrph} \to \mathsf{Psh}(\mathbf{Int})$. We also have a left adjoint $\mathbf{wGrph} \to \mathbf{Cat}/\mathcal{R}$, given by the free category construction, whose functor to $\mathcal{R}$ sends a path to the sum of its weights. The diagram of left adjoints commutes:

$$
\begin{array}{ccc}
\mathbf{Grph} & \longrightarrow & \mathsf{Psh}(\mathbf{Int}_N) \\
\searrow & & \searrow \\
\mathbf{wGrph} & \longrightarrow & \mathsf{Psh}(\mathbf{Int}) \\
\downarrow & \downarrow & \downarrow \\
\mathbf{Cat}/\mathcal{N} & \longrightarrow \mathbf{DCF}/\mathcal{N} \simeq \widetilde{\mathbf{Int}}_N & \\
\searrow & & \searrow \\
\mathbf{Cat}/\mathcal{R} & \longrightarrow & \mathbf{DCF}/\mathcal{R} \simeq \widetilde{\mathbf{Int}}
\end{array}
$$

A.3. **The Conduché perspective on interval sheaves and machines.** The equivalence (53) between interval sheaves and discrete Conduché fibrations is in particular expressed as follows. To every $\mathbf{Int}$-sheaf A, we may associate a category $\overline{A}$ called its *associated category*, as well as a functor $\mathsf{len}\colon \overline{A} \to \mathcal{R}$ called its *length* functor, similarly to (52). Explicitly, the object set of $\overline{A}$ is the set $\mathrm{ob}\,\overline{A} \coloneqq A(0)$ of germs in A; morphisms in $\overline{A}$ are sections $a \in A(\ell)$ of arbitrary length; composition is given by gluing sections. The functor len assigns to each morphism a its length $\mathsf{len}(a) \coloneqq \ell$. Moreover, sheaf morphisms $F\colon A \to B$ correspond to length-preserving functors $\overline{F}\colon \overline{A} \to \overline{B}$ over $\mathcal{R}$.

Under the above correspondence, we can view continuous machines, Definition 4.1.1, as

$$
\mathsf{Mch}(A,B) = \widetilde{\mathbf{Int}}/(A \times B) \simeq \left(\mathbf{DCF}/\mathcal{R}\right) \Big/ \left(\overline{A \times B} \xrightarrow{\mathsf{len}} \mathcal{R}\right) \cong \mathbf{DCF}/\overline{A} \times \overline{B},
$$

namely themselves as discrete Conduché fibrations over the product of the associated categories of the input and output interval sheaves.

Moreover, the notions of totality and determinism for sheaf morphisms defined by Proposition 4.2.1 also have equivalent expressions in the language of discrete Conduché fibrations. Let $p\colon S \to A$ be a sheaf morphism, let $\overline{p}\colon \overline{S} \to \overline{A}$ be the associated functor and s the source map. Then p is total (resp. deterministic) if and only if the induced function h

$$
\begin{array}{ccc}
\overline{S}_1 & \xrightarrow{\quad s \quad} & \\
 & \searrow h & \\
\overline{p}_1 \Big\downarrow & S' \longrightarrow & \overline{S}_0 \\
 & \downarrow \quad {}^{\lrcorner} & \Big\downarrow \overline{p}_0 \\
 & \overline{A}_1 \xrightarrow{\ s\ } & \overline{A}_0
\end{array}
$$

is surjective (resp. injective); this is precisely condition (2).

Notice that the above conditions of $\overline{p}$ could in fact be defined for an arbitrary functor $F\colon \mathcal{C} \to \mathcal{D}$. In the spirit of Definition A.1.1, one could define F to be a (discrete) *epi-opfibration* (resp. *mono-opfibration*) if $h\colon \mathcal{C}_1 \to \mathcal{D}_1 \times_{\mathcal{D}_0} \mathcal{C}_0$ is surjective (resp. injective). These express whether for each morphism in the base category and object above, say, the target, there exists at least one, or maximum one, appropriate lifting in the domain category. In the case h is bijective, we recover the notion of a discrete opfibration.

Thus, p is total (resp. deterministic) in the sense of Definition 4.2.2 if and only if $\overline{p}$ is an epi-opfibration (resp. mono-opfibration). More informally, if and only if for all functors

a, b as shown in the diagram on the left (resp. right), there exists a dotted lift:
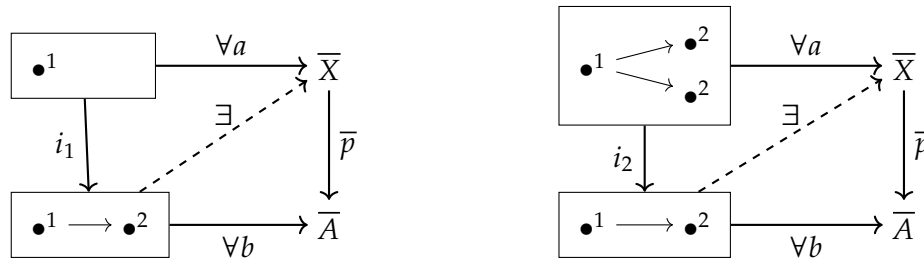
where i_1 and i_2 are the obvious functors preserving object labels. Finally, p is total and deterministic if and only if $\overline{p}$ is a discrete opfibration.

References

[ABLR02] Jiří Adámek, Francis Borceux, Stephen Lack, and Jiří Rosický. A classification of accessible categories. *J. Pure Appl. Algebra*, 175(1-3):7–30, 2002. Special volume celebrating the 70th birthday of Professor Max Kelly.

[ABP99] Samson Abramsky, Richard Blute, and Prakash Panangaden. Nuclear and trace ideals in tensored ∗-categories. *J. Pure Appl. Algebra*, 143(1-3):3–47, 1999. Special volume on the occasion of the 60th birthday of Professor Michael Barr (Montreal, QC, 1997).

[Age96] Pierre Ageron. Effective taxonomies and crossed taxonomies. *Cahiers Topologie Géom. Différentielle Catég.*, 37(2):82–90, 1996.

[Bar71] Michael Barr. Exact categories. In *Exact categories and categories of sheaves*, pages 1–120. Springer, 1971.

[Ber02] Clemens Berger. A cellular nerve for higher categories. *Advances in Mathematics*, 169:118–175, 2002.

[BF00] Marta Bunge and Marcelo P. Fiore. Unique factorisation lifting functors and categories of linearly-controlled processes. *Mathematical Structures in Computer Science*, 10(2):137–163, 2000.

[BMW12] Clemens Berger, Paul-André Melliès, and Mark Weber. Monads with arities and their associated theories. *Journal of Pure and Applied Algebra*, 216(8):2029–2048, 2012.

[DS97] Brian Day and Ross Street. Monoidal bicategories and Hopf algebroids. *Adv. Math.*, 129(1):99–157, 1997.

[Fio00] Marcelo P. Fiore. Fibred models of processes: Discrete, continuous, and hybrid systems. In *Proceedings of the International Conference IFIP on Theoretical Computer Science, Exploring New Frontiers of Theoretical Informatics*, TCS '00, pages 457–473, Berlin, Heidelberg, 2000. Springer-Verlag.

[Fon16] Brendan Fong. *The Algebra of Open and Interconnected Systems*. PhD thesis, University of Oxford, 2016.

[FS18] Brendan Fong and David Spivak. Hypergraph Categories. arXiv:1806.08304 [math.CT], 2018.

[Hed18] Jules Hedges. Morphisms of open games. *Electronic Notes in Theoretical Computer Science*, 341:151 – 177, 2018. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).

[Her00] Claudio Hermida. Representable multicategories. *Adv. Math.*, 151(2):164–225, 2000.

[Jac99] Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

[JNW96] Andrĩǐ Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164 – 185, 1996.

[Joh99] Peter Johnstone. A note on discrete Conduché fibrations. *Theory Appl. Categ.*, 5:No. 1, 1–11, 1999.

[Joh02] Peter T. Johnstone. *Sketches of an elephant: a topos theory compendium.*, volume 43 of *Oxford Logic Guides*. The Clarendon Press Oxford University Press, New York, 2002.

[JS93] André Joyal and Ross Street. Braided tensor categories. *Adv. Math.*, 102(1):20–78, 1993.

[JSV96] André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996.

[Koc11]   Joachim Kock. Polynomial functors and trees. *International Mathematics Research Notices*, 2011(3):609–673, 2011.

[KSW00]   P. Katis, N. Sabadini, and R.F.C. Walters. On the algebra of systems with feedback and boundary. *Rendiconti del Circolo Matematico di Palermo Serie II*, 63:123–156, 2000.

[Lam69]   Joachim Lambek. Deductive systems and categories. II. Standard constructions and closed categories. In *Category Theory, Homology Theory and their Applications, I (Battelle Institute Conference, Seattle, Wash., 1968, Vol. One)*, pages 76–122. Springer, Berlin, 1969.

[Law86]   F. William Lawvere. State categories and response functors. Unpublished manuscript, 1986.

[Lei04a] Tom Leinster. *Higher operads, higher categories*. Number 298 in London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 2004.

[Lei04b] Tom Leinster. Nerves of algebras. Talk at CT04, Vancouver, 2004.

[LS17]   Edward A. Lee and Sanjit A. Seshia. *Introduction to Embedded Systems, A Cyber-Physical Systems Approach*. MIT Press, 2017. Second Edition.

[Mil96]   Robin Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, Nov 1996.

[MMS09]   M. Markl, S. Merkulov, and S. Shadrin. Wheeled props, graph complexes and the master equation. *Journal of Pure and Applied Algebra*, 213(4):496 – 535, 2009.

[MV18]   Joe Moeller and Christina Vasilakopoulou. Monoidal grothendieck construction, 2018.

[Pai90]   Valeria Correa Vaz De Paiva. *The Dialectica Categories*. PhD thesis, University of Cambridge, UK, 1990.

[RS13]   Dylan Rupel and David I. Spivak. The operad of temporal wiring diagrams: formalizing a graphical language for discrete-time processes, 2013.

[Sel97]   Peter Selinger. First-order axioms for asynchrony. In *International Conference on Concurrency Theory*, pages 376–390. Springer, 1997.

[Spi13]   David I. Spivak. The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits. *CoRR*, abs/1305.0297, 2013.

[Spi15]   David I Spivak. The steady states of coupled dynamical systems compose according to matrix arithmetic. *arXiv preprint: 1512.00802*, 2015.

[SS17]   Patrick Schultz and David I. Spivak. Temporal type theory: A topos-theoretic approach to systems and behavior. 2017.

[SSR16]   David I. Spivak, Patrick Schultz, and Dylan Rupel. String diagrams for traced and compact categories are oriented 1-cobordisms. 2016.

[Sta16]   Michael Stay. Compact closed bicategories. *Theory and Applications of Categories*, 31:755–798, 2016.

[SY13]   Janos Sztipanovits and Susan Ying. Strategic r&d opportunities for 21st century cyber-physical systems. Technical report, National Institute of Standards and Technology, 2013.

[VSL15]   Dmitry Vagner, David I. Spivak, and Eugene Lerman. Algebras of open dynamical systems on the operad of wiring diagrams. *Theory Appl. Categ.*, 30:Paper No. 51, 1793–1822, 2015.

[Web07]   Mark Weber. Familial 2-functors and parametric right adjoints. *Theory and Applications of Categories*, 18(22):665–732, 2007.

[Win87]   Glynn Winskel. *Event structures*, pages 325–392. Springer Berlin Heidelberg, Berlin, Heidelberg, 1987.

[WP13]   Jan C Willems and Jan W Polderman. *Introduction to mathematical systems theory: a behavioral approach*, volume 26. Springer Science & Business Media, 2013.