ALGEBRAS OF OPEN DYNAMICAL SYSTEMS ON THE OPERAD OF WIRING DIAGRAMS

DMITRY VAGNER, DAVID I. SPIVAK, AND EUGENE LERMAN

Abstract. In this paper, we use the language of operads to study open dynamical systems. More specifically, we study the algebraic nature of assembling complex dynamical systems from an interconnection of simpler ones. The syntactic architecture of such interconnections is encoded using the visual language of wiring diagrams. We define the symmetric monoidal category $\mathbf{W}$, from which we may construct an operad $\mathcal{O}_W$, whose objects are black boxes with input and output ports, and whose morphisms are wiring diagrams, thus prescribing the algebraic rules for interconnection. We then define two $\mathbf{W}$-algebras $\mathcal{G}$ and $\mathcal{L}$, which associate semantic content to the structures in $\mathbf{W}$. Respectively, they correspond to general and to linear systems of differential equations, in which an internal state is controlled by inputs and produces outputs. As an example, we use these algebras to formalize the classical problem of systems of tanks interconnected by pipes, and hence make explicit the algebraic relationship between systems at different levels of granularity.

Contents

1. Introduction 1
2. Preliminary Notions 4
3. The Operad of Wiring Diagrams 8
4. The Algebra of Open Systems 14
5. The Subalgebra of Linear Open Systems 16
References 19

1. Introduction

This paper uses diagrammatic language to understand how dynamical systems that describe processes can be built up from the systems that describe its subprocesses. More precisely, we will define a symmetric monoidal category $\mathbf{W}$ of black boxes and wiring diagrams. Its underlying operad $\mathcal{O}_W$ is a graphical language for building larger black boxes out of an interconnected set of smaller ones. We then define two $\mathbf{W}$-algebras, $\mathcal{G}$ and $\mathcal{L}$, which encode open dynamical systems,

Spivak acknowledges support from ONR grant N000141310260 and AFOSR grant FA9550-14-1-0031.
i.e., differential equations of the form

\[
\begin{align*}
\dot{Q} &= f^{in}(Q, \text{input}) \\
\text{output} &= f^{out}(Q)
\end{align*}
\]

(1)

where \(Q\) represents an internal state vector, \(\dot{Q} = \frac{dQ}{dt}\) is its time derivative, and \(\text{input}\) and \(\text{output}\) represent inputs to and outputs from the system. In \(\mathcal{G}\), the functions \(f^{in}\) and \(f^{out}\) are smooth, whereas in the subalgebra \(\mathcal{L} \subseteq \mathcal{G}\), they are moreover linear. The fact that \(\mathcal{G}\) and \(\mathcal{L}\) are \(\mathcal{W}\)-algebras is capturing the fact that these systems are closed under wiring diagram interconnection.

Our notion of interconnection is a generalization of that in Deville and Lerman \([\text{DL10}], [\text{DL12}], [\text{DL14}]\) Their version of interconnection produces a closed system from open ones, and can be understood in the present context as a morphism whose codomain is the closed box (see Definition 3.8).

The present work exists within a broader movement of using the visual language of diagrams and networks to study systems of various sorts. Category theory serves as an organizational framework that coheres the various visual languages used in disparate applications. It is demonstrated in \([\text{BS11}]\) and \([\text{Coe13}]\) that one can study applications in diverse fields such as physics, topology, logic, linguistics, and computation using the language of monoidal categories. More recently, as in \([\text{BB12}]\), there has been growing interest in viewing more traditionally applied fields, such as ecology, biology, chemistry, and electrical engineering, through such a lens. Specifically, category theory has been used to draw connections among visual languages such as Feynman diagrams, circuit diagrams, social networks, Petri nets, flow charts, and planar knot diagrams. This research is building toward what some would consider a new way to organize basic techniques commonly employed in applied mathematics literature.

Joyal and Street’s work on string diagrams \([\text{JS91}]\) and (with Verity) on traced monoidal categories \([\text{JSV96}]\) has been used for decades to visualize compositions and feedback in networked systems. Traced monoidal categories are a general framework for systems that have shown up, for example, in recent developments in the theory of flow charts \([\text{AMMO10}]\). Any traced monoidal category can be viewed as an algebra on our monoidal category \(\mathcal{W}\) of wiring diagrams, though that will not be explained here (see \([\text{RS15}]\)). Flow diagrams, a precursor to string diagrams, have been used in the mathematical theory of computation since the 1970’s \([\text{Sco71}]\). The main addition of the present work is the inclusion of an outer box, which allows for holarchic \([\text{Koe67}]\) combinations of these diagrams. That is, the parts can be assembled into a whole which can itself be a part. The composition of such assemblies can now be viewed as morphism composition in an operad.

This paper is the third in a series, following \([\text{RS13}]\) and \([\text{Spi13}]\), on using wiring diagrams to model interactions. Here we present a distinct algebra, that of open systems, to the algebras of relations and of propagators studied in earlier works. Beyond the dichotomy of discrete vs. continuous, these algebras are markedly different in structure. For one thing, the internal wires in \([\text{RS13}]\) themselves carry state, whereas here, a wire should be thought of as instantaneously transmitting its contents from an output site to an input site. Another difference between our algebra and those of previous works is that the algebras here involve open systems in which, as in (1), the instantaneous change of state is a function of the current
state and the input, whereas the output depends only on the current state (see Definition 4.2).

1.1. Motivating example. The motivating example for the algebras in this paper comes from classical differential equations pedagogy; namely, systems of tanks containing salt water concentrations, with pipes carrying fluid among them. The systems of ODEs produced by such applications constitute a subset of those our language can address; they are linear systems with a certain form (see Example 5.3). To ground the discussion, we begin by considering a specific example.

Example 1.1. Figure 1 below is a wiring diagram version of a problem found in Boyce and DiPrima’s canonical text [BD65, Figure 7.1.6].

![Wiring Diagram for Example 1.1](image)

**Figure 1.** A wiring diagram $\Phi: X_1, X_2 \to Y$ in $\mathcal{OW}$.

In this diagram, $X_1$ and $X_2$ are boxes that represent tanks consisting of salt water solution. The functions $Q_1(t)$ and $Q_2(t)$ represent the amount of salt (in ounces) found in 30 and 20 gallons of water, respectively. These tanks are interconnected with each other by pipes embedded within a system $Y$. The prescription for how wires are attached among the boxes is formally encoded in the wiring diagram $\Phi: X_1, X_2 \to Y$, as we will discuss in Definition 3.5.

Both tanks are being fed salt water concentrations at constant rates from the outside world. Specifically, $X_1$ is fed a 1 ounce salt per gallon water solution at 1.5 gallons per minute and $X_2$ is fed a 3 ounce salt per gallon water solution at 1 gallon per minute. The tanks also both feed each other their solutions, with $X_1$ feeding $X_2$ at 3 gallons per minute and $X_2$ feeding $X_1$ at 1.5 gallons per minute. Finally, $X_2$ feeds the outside world its solution at 2.5 gallons per minute.

The dynamics of the salt water concentrations both within and leaving each tank $X_i$ is encoded in a linear open system $f_i$, consisting of a differential equation for $Q_i$ and a readout map for each $X_i$ output (see Definition 2.9). Our algebra $\mathcal{L}$ allows one to assign a linear open system $f_i$ to each tank $X_i$ and, using $\Phi: X_1, X_2 \to Y$, a linear open system is functorially assigned to box $Y$. This paper will explore this construction in detail, in particular providing explicit formulas for it in both the linear case, as well as for more general systems of ODEs.
2. Preliminary Notions

Throughout this paper we use the language of monoidal categories and functors. Depending on the audience, appropriate background on basic category theory can be found in MacLane [ML98], Awodey [Awo10], or Spivak [Spi14]. Leinster [Lei04] is a good source for more specific information on monoidal categories and operads. We refer the reader to [KFA69] for an introduction to dynamical systems.

Notation. We denote the category of sets and functions by \( \text{Set} \) and the full subcategory spanned by finite sets as \( \text{FinSet} \). We generally do not concern ourselves with cardinality issues. We follow Leinster [Lei04] and use \( \times \) for binary product and \( \Pi \) for arbitrary product, and dually \( + \) for binary coproduct and \( \bigoplus \) for arbitrary coproduct in any category. By \textit{operad} we always mean symmetric colored operad or, equivalently, symmetric multicategory.

2.1. Monoidal categories and operads. In Section 3, we will construct the symmetric monoidal category \( (W, \oplus, 0) \) of boxes and wiring diagrams, which we often simply denote as \( W \). We will sometimes consider the underlying operad \( O_W \), obtained by applying the canonical functor \( O : \text{SMC} \to \text{Opd} \) to \( W \). A brief description of this functor \( O \) is given below in Definition 2.1.

Definition 2.1. Let \( \text{SMC} \) denote the category of symmetric monoidal categories and lax monoidal functors; and \( \text{Opd} \) be the category of operads and operad functors. Given a symmetric monoidal category \( (C, \otimes, 1_C) \in \text{Ob SMC} \), we define the operad \( OC \) as follows:

\[
\text{Ob } OC := \text{Ob } C, \quad \text{Hom}_{OC}(X_1, \ldots, X_n; Y) := \text{Hom}_C(X_1 \otimes \cdots \otimes X_n, Y)
\]

for any \( n \in \mathbb{N} \) and objects \( X_1, \ldots, X_n, Y \in \text{Ob } C \).

Now suppose \( F : (C, \otimes, 1_C) \to (D, \odot, 1_D) \) is a lax monoidal functor in \( \text{SMC} \). By definition such a functor is equipped with a morphism

\[
\mu : FX_1 \odot \cdots \odot FX_n \to F(X_1 \otimes \cdots \otimes X_n),
\]

natural in the \( X_i \), called the coherence map. With this map in hand, we define the operad functor \( OF : OC \to OD \) by stating how it acts on objects \( X \) and morphisms \( \Phi : X_1, \ldots, X_n \to Y \) in \( OC \):

\[
OF(X) := F(X), \quad OF(\Phi : X_1, \ldots, X_n \to Y) := F(\Phi) \circ \mu : FX_1 \odot \cdots \odot FX_n \to FY.
\]

Example 2.2. Consider the symmetric monoidal category \( (\mathbb{S}et, \times, *) \), where \( \times \) is the cartesian product of sets and \( * \) a one element set. Define \( \text{Sets} := O\text{Set} \) as in Definition 2.1. Explicitly, \( \text{Sets} \) is the operad in which an object is a set and a morphism \( f : X_1, \ldots, X_n \to Y \) is a function \( f : X_1 \times \cdots \times X_n \to Y \).

Definition 2.3. Let \( C \) be a symmetric monoidal category and let \( \text{Set} = (\mathbb{S}et, \times, *) \) be as in Example 2.2. A \( C \)-algebra is a lax monoidal functor \( C \to \text{Set} \). Similarly, if \( D \) is an operad, a \( D \)-algebra is defined as an operad functor \( D \to \text{Sets} \).

To avoid subscripts, we will generally use the formalism of SMCs in this paper. Definition 2.1 can be applied throughout to recast everything we do in terms of operads. The primary reason operads may be preferable in applications is that they suggest more compelling pictures. Hence throughout this paper, depictions
of wiring diagrams will usually be operadic, i.e. have many input boxes wired together into one output box.

2.2. Typed sets. Each box in a wiring diagram will consist of finite sets of ports, each labelled by a type. To capture this idea precisely, we define the notion of typed finite sets. Recall that a cartesian category is a category that is closed under taking finite products.

Definition 2.4. Let \( C \) be a cartesian category. The category of \( C \)-typed finite sets, denoted \( \text{TFS}_C \), is defined as follows. An object in \( \text{TFS}_C \) is a finite set of \( C \)-objects,

\[
\text{Ob}\ \text{TFS}_C := \{(A, \tau) \mid A \in \text{Ob}\ \text{FinSet}, \tau : A \to \text{Ob}\ C\}.
\]

For any element \( a \in A \), we call the object \( \tau(a) \) its type. If the typing function \( \tau \) is clear from context, we may denote \( (A, \tau) \) simply by \( A \).

A morphism \( q : (A, \tau) \to (A', \tau') \) in \( \text{TFS}_C \) consists of a function \( q : A \to A' \) that makes the following diagram of finite sets commute:

\[
\begin{array}{ccc}
A & \xrightarrow{q} & A' \\
\downarrow{\tau} & & \downarrow{\tau'} \\
\text{Ob}\ C & & \\
\end{array}
\]

We refer to the morphisms of \( \text{TFS}_C \) as \( C \)-typed functions. If a \( C \)-typed function \( q \) is bijective, we call it a \( C \)-typed bijection.

For any category \( C \), the category \( \text{TFS}_C \) is isomorphic to the slice category \( \text{FinSet}/_C \). Note that \( \text{TFS}_C \) is closed under taking finite coproducts.

Definition 2.5. Let \( C \) be a cartesian category, and let \( (A, \tau) \in \text{Ob}\ \text{TFS}_C \) be a \( C \)-typed finite set. Its \textit{dependent product} \( \text{Proj}(A, \tau) \in \text{Ob}\ C \) is defined as

\[
\text{Proj}(A, \tau) := \prod_{a \in A} \tau(a).
\]

Given a typed function \( q : (A, \tau) \to (A', \tau') \) in \( \text{TFS}_C \) we define

\[
\text{Proj}(A', \tau') 
\]

to be the unique morphism for which the following diagram commutes for all \( a \in A \):

\[
\begin{array}{ccc}
\prod_{a' \in A'} \tau'(a') & \xrightarrow{\text{Proj}} & \prod_{a \in A} \tau(a) \\
\text{Proj}(q(a)) \downarrow & & \downarrow \text{Proj}(\tau(a)) \\
\tau(q(a)) & = & \tau(a)
\end{array}
\]

By the universal property for products, this defines a functor,

\[
\text{Proj} : \text{TFS}_C^{op} \to C.
\]

Lemma 2.6. The \textit{dependent product} functor sends coproducts in \( \text{TFS}_C \) to products in \( C \). That is, for any finite set \( I \) whose elements index typed finite sets \( (A_i, \tau_i) \), there is a canonical isomorphism in \( C \),

\[
\prod_{i \in I} (A_i, \tau_i) \cong \prod_{i \in I} (A_i, \tau_i).
\]
Proof. This is straightforward. □

Remark 2.7. The category of second-countable smooth manifolds and smooth maps is essentially small (by the embedding theorem) so we choose a small representative and denote it Man. Note that Man is cartesian. Manifolds will be our default typing, in the sense that we generally take \( C := \text{Man} \) in Definition 2.4 and denote

\[ \text{TFS} := \text{TFS}_{\text{Man}}. \]

We thus refer to the objects, morphisms, and isomorphisms in TFS as typed finite sets, typed functions, and typed bijections, respectively.

Remark 2.8. The ports of each box in a wiring diagram will be labeled by manifolds because they are the natural setting for geometrically interpreting differential equations (see [Spi65]). For simplicity, one may wish to restrict attention to the full subcategory \( \text{Euc} \) of Euclidean spaces \( \mathbb{R}^n \) for \( n \in \mathbb{N} \), because they are the usual domains for ODEs found in the literature; or to the (non-full) subcategory \( \text{Lin} \) of Euclidean spaces and linear maps between them, because they characterize linear systems of ODEs. We will return to \( \text{TFS}_{\text{Lin}} \) in Section 5.

2.3. Open systems. As a final preliminary, we define our notion of open dynamical system. Recall that every manifold \( M \) has a tangent bundle manifold, denoted \( TM \), and a smooth projection map \( p: TM \to M \). For any point \( m \in M \), the preimage \( T_m M := p^{-1}(m) \) has the structure of a vector space, called the tangent space of \( M \) at \( m \). If \( M \cong \mathbb{R}^n \) is a Euclidean space then also \( T_m M \cong \mathbb{R}^n \) for every point \( m \in M \). A vector field on \( M \) is a smooth map \( g: M \to TM \) such that \( p \circ g = \text{id}_M \). See [Spi65] or [War83] for more background.

For the purposes of this paper we make the following definition of open systems; this may not be completely standard.

Definition 2.9. Let \( M, U_{\text{in}}, U_{\text{out}} \in \text{Ob Man} \) be smooth manifolds and \( TM \) be the tangent bundle of \( M \). Let \( f = (f_{\text{in}}, f_{\text{out}}) \) denote a pair of smooth maps

\[
\begin{align*}
& f_{\text{in}}: M \times U_{\text{in}} \to TM \\
& f_{\text{out}}: M \to U_{\text{out}}
\end{align*}
\]

where, for all \((m, u) \in M \times U_{\text{in}}\) we have \( f_{\text{in}}(m, u) \in T_m M\); that is, the following diagram commutes:

\[ M \times U_{\text{in}} \xrightarrow{f_{\text{in}}} TM \]

We sometimes use \( f \) to denote the whole tuple,

\[ f = (M, U_{\text{in}}, U_{\text{out}}, f), \]

which we refer to as an open dynamical system (or open system for short). We call \( M \) the state space, \( f_{\text{in}} \) the differential equation, and \( f_{\text{out}} \) the readout map of the open system.

Note that the pair \( f = (f_{\text{in}}, f_{\text{out}}) \) is determined by a single smooth map

\[ f: M \times U_{\text{in}} \to TM \times U_{\text{out}}, \]
which, by a minor abuse of notation, we also denote by $f$.

In the special case that $M, U^{\text{in}}, U^{\text{out}} \in \text{Ob Lin}$ are Euclidean spaces and $f$ is a linear map (or equivalently $f^{\text{in}}$ and $f^{\text{out}}$ are linear), we call $f$ a linear open system.

Remark 2.10. In practice, open system typically occur in the form of equations such as

$$\dot{m} = f^{\text{in}}(m, u^{\text{in}})$$

$$u^{\text{out}} = f^{\text{out}}(m)$$

where $m \in M, u^{\text{in}} \in U^{\text{in}}, u^{\text{out}} \in U^{\text{out}}$, as seen earlier in (1).

Example 2.11. We give a few special cases to fix ideas. Let $M$ be a smooth manifold, and let $U^{\text{in}} = U^{\text{out}} = \mathbb{R}^0$ be trivial. Then an open system in the sense of Definition 2.9 is a smooth map $f : M \rightarrow TM$ over $M$, in other words, a vector field on $M$. From the geometric point of view, vector fields are autonomous dynamical systems; see [Tes12].

More generally, for an arbitrary manifold $U^{\text{in}}$, a map $M \times U^{\text{in}} \rightarrow TM$ can be considered as a function $U^{\text{in}} \rightarrow \text{VF}(M)$, where $\text{VF}(M)$ is the set of vector fields on $M$. Hence, $U^{\text{in}}$ controls the behavior of the system in the usual sense.

By defining the appropriate morphisms, we can consider open dynamical systems as being objects in a category. We are not aware of this notion being defined previously in the literature, but it is convenient for our purposes.

Definition 2.12. Suppose that $M_i, U^{\text{in}}_i, U^{\text{out}}_i \in \text{Ob Man}$ and $(M_i, U^{\text{in}}_i, U^{\text{out}}_i, f_i)$ is an open system for each $i \in \{1, 2\}$. A morphism of open systems

$$\zeta : (M_1, U^{\text{in}}_1, U^{\text{out}}_1, f_1) \rightarrow (M_2, U^{\text{in}}_2, U^{\text{out}}_2, f_2)$$

is a triple $(\zeta_M, \zeta_{U^{\text{in}}_1}, \zeta_{U^{\text{out}}_1})$ of smooth maps $\zeta_M : M_1 \rightarrow M_2, \zeta_{U^{\text{in}}_1} : U^{\text{in}}_1 \rightarrow U^{\text{in}}_2$, and $\zeta_{U^{\text{out}}_1} : U^{\text{out}}_1 \rightarrow U^{\text{out}}_2$, such that the following diagram commutes:

\[
\begin{array}{ccc}
M_1 \times U^{\text{in}}_1 & \xrightarrow{f_1} & TM_1 \times U^{\text{out}}_1 \\
\zeta_M \times \zeta_{U^{\text{in}}_1} \downarrow & & \downarrow T\zeta_M \times \zeta_{U^{\text{out}}_1} \\
M_2 \times U^{\text{in}}_2 & \xrightarrow{f_2} & TM_2 \times U^{\text{out}}_2 \\
\end{array}
\]

This defines the category $\text{ODS}$ of open dynamical systems. We define the subcategory $\text{ODS}_{\text{Lin}} \subseteq \text{ODS}$ by restricting our objects to linear open systems, as in Definition 2.9, and imposing that $\zeta$ consist entirely of linear maps.

Lemma 2.13. The category $\text{ODS}$ of open systems has all finite products. That is, if $I$ is a finite set and $f_i = (M_i, U^{\text{in}}_{i, i}, U^{\text{out}}_{i, i}, f_{i, i}) \in \text{Ob ODS}$ is an open system for each $i \in I$, then their product is

$$\prod_{i \in I} f_i = \left( \prod_{i \in I} M_i, \prod_{i \in I} U^{\text{in}}_{i, i}, \prod_{i \in I} U^{\text{out}}_{i, i}, \prod_{i \in I} f_{i, i} \right)$$

with the obvious projection maps.

Proof. This is straightforward. □
3. The Operad of Wiring Diagrams

In this section, we define the symmetric monoidal category \((W, \oplus, 0)\) of wiring diagrams, of which \(\mathcal{O}_W\) is the associated operad (see Definition 2.1). We begin by defining the objects of \(W\), which we call black boxes, or simply boxes.

**Definition 3.1.** A box \(X\) is an ordered pair of \(\text{Man}\)-typed finite sets, \(X = (X^{\text{in}}, X^{\text{out}}) \in \text{Ob}\ TFS \times \text{Ob}\ TFS\).

Let \(X^{\text{in}} = (A, \tau)\) and \(X^{\text{out}} = (A', \tau')\). Then we refer to elements \(a \in A\) and \(a' \in A'\) as input ports and output ports, respectively. We call \(\tau(a) \in \text{Ob}\ \text{Man}\) the type of port \(a\), and similarly for \(\tau'(a')\).

**Remark 3.2.** For any cartesian category \(C\), we may define the symmetric monoidal category \(W_C\) by replacing \(\text{Man}\) by \(C\), and \(\text{TFS}\) with \(\text{TFS}_C\), in Definition 3.1. In particular, as in Remark 2.8, we have the symmetric monoidal category \(W_{\text{Lin}}\) of linearly typed wiring diagrams.

What we are calling a box is nothing more than an interface; at this stage it has no semantics, e.g. in terms of differential equations. Each box can be given a pictorial representation, as in Example 3.3.

**Example 3.3.** In Figure 2, we depict a box \(X = (\{a, b\}, \{c\})\), with both input ports having \(\mathbb{R}\) as their type, and the output port having \(\mathbb{R}^3\) as its type. As a convention, input and output ports will connect on the left and right sides of the box, respectively.

![Figure 2. A box with two input ports, both with type \(\mathbb{R}\), and one output port, with type \(\mathbb{R}^3\).](image)

**Example 3.4.** For a concrete example on how to go the other way—that is, from pictures to formalism—consider the boxes in Figure 1. Observing the set of ports attached to the left and right side of each box, we see \(X_1 = (\{X^{\text{in}}_{1a}, X^{\text{in}}_{1b}\}, \{X^{\text{out}}_{1a}\})\), \(X_2 = (\{X^{\text{in}}_{2a}, X^{\text{in}}_{2b}\}, \{X^{\text{out}}_{2a}, X^{\text{out}}_{2b}\})\), and \(Y = (\{Y^{\text{in}}_{a}, Y^{\text{in}}_{b}\}, \{Y^{\text{out}}_{a}\})\). Each of the ports has type \(\mathbb{R}\), denoting the rate of salt being carried as a real number of ounces per minute. We will see in Remark 3.6 that if a port connects two boxes, the associated types must be the same.

Now that we have specified the objects of \(W\), we can define the morphisms. The following definition is a bit terse, but we will unpack it afterwards.

**Definition 3.5.** Let \(X, Y \in \text{Ob}\ W\). Then a wiring diagram \(\Phi: X \to Y\) is a typed bijection (see Definition 2.4)

\[
\varphi: X^{\text{in}} + Y^{\text{out}} \xrightarrow{\cong} X^{\text{out}} + Y^{\text{in}},
\]

satisfying the following condition:
no passing wires: \( \varphi(Y^{\text{out}}) \cap Y^{\text{in}} = \emptyset \), or equivalently \( \varphi(Y^{\text{out}}) \subseteq X^{\text{out}} \).

We often identify the morphism \( \Phi \) with the typed bijection \( \varphi \).

By a wire in \( \Phi \), we mean a pair \((a, b)\), where \( a \in X^{\text{in}} + Y^{\text{out}} \), \( b \in X^{\text{out}} + Y^{\text{in}} \), and \( \varphi(a) = b \). In other words a wire in \( \Phi \) is a pair of ports connected by \( \Phi \).

**Remark 3.6.** The definition of wiring diagrams includes various conditions on the function \( \varphi \) as in (3). The condition that \( \varphi \) be typed, as in Definition 2.4, ensures that if two ports are connected by a wire then the associated types are the same, as we commented in Example 3.4. The condition that \( \varphi \) be bijective prohibits exposed ports and split ports, depicted in Figure 3a, by imposing surjectivity and injectivity, respectively. Finally, the no passing wires condition on \( \varphi(Y^{\text{out}}) \) prohibits wires that go straight across the \( Y \) box, as seen in the intermediate box of Figure 3b. This condition allows us to avoid mildly pathological compositions such as closed loops.

![Figure 3](image-url)

**Figure 3.** (a) A faux-wiring diagram prohibited by Definition 3.5 because the corresponding typed function \( \varphi \) violates the bijectivity requirement. (b) A composition of diagrams in which a loop emerges because the inner diagram has a (prohibited) passing wire.

The bijectivity and no passing wires conditions in Definition 3.5 are not strictly necessary conditions, but they are imposed because they greatly reduce the complexity of the mathematical formulas.

**Remark 3.7.** Let \( \Phi: X \to Y \) be a wiring diagram, and \( \varphi: X^{\text{in}} + Y^{\text{out}} \to X^{\text{out}} + Y^{\text{in}} \) be the corresponding typed bijection. Under the no passing wires condition, we see that \( \varphi \) can be decomposed into two maps

\[
\begin{align*}
\varphi^{\text{in}} : & X^{\text{in}} \to X^{\text{out}} + Y^{\text{in}} \\
\varphi^{\text{out}} : & Y^{\text{out}} \to X^{\text{out}}
\end{align*}
\]

Suppose we denote the image of \( \varphi^{\text{out}} \) as \( X^{\text{exp}}_{\varphi} \subseteq X^{\text{out}} \), the exports, and its complement as \( X^{\text{loc}}_{\varphi} \), the local ports. Then we can identify \( \Phi \) with the following:

- a subset \( X^{\text{exp}}_{\varphi} \subseteq X^{\text{out}} \), having complement \( X^{\text{loc}}_{\varphi} \), and
- a pair of typed bijections

\[
\begin{align*}
\tilde{\varphi}^{\text{in}} : & X^{\text{in}} \xrightarrow{\cong} X^{\text{loc}}_{\varphi} + Y^{\text{in}} \\
\tilde{\varphi}^{\text{out}} : & Y^{\text{out}} \xrightarrow{\cong} X^{\text{exp}}_{\varphi}
\end{align*}
\]

This description will be used in our proof of Proposition 3.12.

We next define the monoidal product in \( W \).
Definition 3.8. Let \( X_i, Y_i \in \text{Ob} \mathbf{W} \) be boxes and \( \Phi_i : X_i \to Y_i \) be wiring diagrams for \( i \in \{1, 2\} \). The **monoidal product** is a functor \( \oplus : \mathbf{W} \times \mathbf{W} \to \mathbf{W} \) given by
\[
X_1 \oplus X_2 := (X_1^{\text{in}} + X_2^{\text{in}}, X_1^{\text{out}} + X_2^{\text{out}}), \quad \Phi_1 \oplus \Phi_2 := \Phi_1 + \Phi_2.
\]
It is clear that \( \oplus \) is symmetric since the disjoint union of finite sets is symmetric.

Closed boxes will correspond to **autonomous systems**, which do not interact with any outside environment (see Example 2.11).

Example 3.9. As exemplified by Figure 1, we have a conventional way to pictorially represent wiring diagrams \( \Phi : X_1, \ldots, X_n \to Y \) in \( \mathbf{OW} \). Domain boxes \( X_i \) are nested within the codomain box \( Y \), and wires attach various ports to each other via the rules prescribed by the typed bijection \( \varphi \).

Let’s explicitly consider the wiring diagram \( \Phi : X_1, X_2 \to Y \) in Example 1.1; it is a morphism in \( \mathbf{OW} \). By Definition 2.1 we can regard it as a morphism \( \Phi : X \to Y \) in \( \mathbf{W} \), where \( X := X_1 \oplus X_2 \). The values of the corresponding typed bijection \( \varphi : X^{\text{in}} + Y^{\text{out}} \cong X^{\text{out}} + Y^{\text{in}} \) can be read directly from the picture in Figure 1; we record them in Table 1.

| \( w \in X^{\text{in}} + Y^{\text{out}} \) | \( X^{\text{in}}_{1a} \) | \( X^{\text{in}}_{1b} \) | \( X^{\text{in}}_{2a} \) | \( X^{\text{in}}_{2b} \) | \( Y^{\text{out}}_a \) |
| \( \varphi(w) \in X^{\text{out}} + Y^{\text{in}} \) | \( Y^{\text{in}}_b \) | \( X^{\text{out}}_{2b} \) | \( Y^{\text{out}}_a \) | \( X^{\text{out}}_{1a} \) | \( X^{\text{out}}_{2a} \) |

Table 1

To finish defining our category \( \mathbf{W} \) of wiring diagrams, the only missing piece is to define composition of wiring diagrams, which we do in two steps. First, given composable morphisms \( X \xrightarrow{\Phi} Y \xrightarrow{\Psi} Z \) in \( \mathbf{W} \), we provide in Definition 3.10 a typed function which serves as a candidate for their composition. Second, we prove in Proposition 3.12 that it is a valid composition formula, in particular that it satisfies the conditions of Definition 3.5.

Definition 3.10. Let \( \Phi : X \to Y \) and \( \Psi : Y \to Z \) be morphisms in \( \mathbf{W} \), and let \( \varphi \) and \( \psi \) be the corresponding typed bijections. Their **candidate composition** is a typed function
\[
\omega : X^{\text{in}} + Z^{\text{out}} \to X^{\text{out}} + Z^{\text{in}},
\]
defined, using Remark 3.7 (4), as the dashed arrows making the following diagrams commute.

\[
\begin{align*}
X^{\text{in}} & \xrightarrow{\varphi^{\text{in}}} X^{\text{out}} + Y^{\text{in}} \\
\downarrow & \downarrow \varphi^{\text{in}} & \downarrow & \downarrow \varphi^{\text{out}} \\
X^{\text{out}} + Y^{\text{in}} & \xrightarrow{1_{X^{\text{out}}} + \varphi^{\text{out}} + \psi^{\text{out}}} X^{\text{out}} + Y^{\text{out}} + Z^{\text{in}} \\
\varphi^{\text{in}} & \\
X^{\text{out}} & \xrightarrow{\nabla + 1_{Z^{\text{in}}}} X^{\text{out}} + Y^{\text{out}} + Z^{\text{in}} \\
\downarrow & \downarrow \psi^{\text{out}} & \downarrow & \downarrow \varphi^{\text{out}} \\
Z^{\text{out}} & \xrightarrow{1_{Z^{\text{out}}} + \varphi^{\text{out}} + \psi^{\text{out}}} Z^{\text{out}} \\
\downarrow & \downarrow \varphi^{\text{in}} & \downarrow & \downarrow \varphi^{\text{out}} \\
X^{\text{out}} & \xrightarrow{\nabla + 1_{Z^{\text{in}}}} X^{\text{out}} + Y^{\text{out}} + Z^{\text{in}}
\end{align*}
\]

Here \( \nabla : X^{\text{out}} + X^{\text{out}} \to X^{\text{out}} \) is the codiagonal map in \( \mathbf{TFS} \).
In contrast to this seemingly convoluted algebraic definition, pictorial composition of wiring diagrams in $\mathbf{W}$ (or $\mathcal{O}_\mathbf{W}$) is straightforward. As shown below in Figure 4, one nests the inner wiring diagrams into their codomain boxes, which are also the domain boxes for the outer wiring diagram, and then erases these intermediary boxes.

\begin{figure}[h]
\centering
\begin{tabular}{cc}
\includegraphics[width=0.4\textwidth]{figure4a} & \includegraphics[width=0.4\textwidth]{figure4b}
\end{tabular}
\caption{(a) Wiring diagrams $\Psi : Y_1, Y_2 \rightarrow Z$, $\Phi_1 : X_{11} \rightarrow Y_1$, and $\Phi_2 : X_{21}, X_{22} \rightarrow Y_2$ are all drawn in one picture. (b) The composite $\Psi \circ (\Phi_1, \Phi_2) : X_{11}, X_{21}, X_{22} \rightarrow Z$ is shown.}
\end{figure}

\textbf{Example 3.11.} The commutative diagram (5) says that $\omega$ is the composition of four typed functions, which can be interpreted by examining Figures 4a and 4b. Starting with a port in $X^\text{in}$, the function $\varphi$ links it to either a port in $X^\text{out}$, as in the case of $\beta$, or to a port in $Y^\text{in}$, as in the case of $\alpha$. Continuing with $\alpha$, the second step links $\alpha$ to a $Y^\text{out}$ port, and the third step links it to an $X^\text{out}$ port. The fourth step is a simple integration of ports like $\alpha$, which exit some intermediate box $Y$, with ports like $\beta$ that do not.

We now prove that the above data characterizing $\mathbf{W}$ indeed constitutes a symmetric monoidal category, at which point we will have the operad $\mathcal{O}_\mathbf{W}$, as advertised, by applying Definition 2.1.

\textbf{Proposition 3.12.} With objects as defined in 3.1, morphisms in 3.5, monoidal product and identity in 3.8, and composites in 3.10, $(\mathbf{W}, \oplus, 0)$ is a symmetric monoidal category.

\textbf{Proof.} Let $X, X', X'' \in \text{Ob} \mathbf{W}$. We readily observe the following canonical isomorphisms.

\begin{align*}
X \oplus 0 &= X = 0 \oplus X & \text{(unity)} \\
(X \oplus X') \oplus X'' &= X \oplus (X' \oplus X'') & \text{(associativity)} \\
X \oplus X' &= X' \oplus X & \text{(commutativity)}
\end{align*}

Thus the monoidal product $\oplus$ is well behaved on objects and it is similarly easy to show it is functorial. Hence it simply remains to show that $\mathbf{W}$ is a category. Let $X \xrightarrow{\varphi} Y \xrightarrow{\psi} Z$ be in $\mathbf{W}$, and let $\varphi$ and $\psi$ be the associated typed bijections. We first prove that our candidate composition $\omega$, from Definition 3.10, is indeed a wiring diagram in the sense of Definition 3.5. We directly compute that $\omega(Z^\text{out}) = \varphi(\psi(Z^\text{out})) \subseteq \varphi(Y^\text{out}) \subseteq X^\text{out}$, so it satisfies the no passing wires condition. We need to check that $\omega$ is a bijection.
Recall the exports, $X^\exp_\varphi = \varphi(Y^\out)$, local ports, $X^\loc_\varphi = X^\out \setminus X^\exp_\varphi$, and typed bijections

$$
\begin{align*}
\varphi^\in : & \quad X^\in \xrightarrow{\cong} X^\loc_\varphi + Y^\in \\
\varphi^\out : & \quad Y^\out \xrightarrow{\cong} X^\exp_\varphi
\end{align*}
$$

from Remark 3.7, as well as the analogous isomorphisms $\tilde{\psi}^\in, \tilde{\psi}^\out$ for $\psi$. We can use these decompositions to recast our definition (5) of $\omega$ in one commutative diagram of typed finite sets.

$$
\begin{array}{c}
X^\in + Z^\out \xrightarrow{\cong} X^\out + Z^\in \\
X^\loc_\varphi + Y^\in + Y^\exp \xrightarrow{\cong} X^\loc_\varphi + X^\exp + Z^\in \\
I_{X^\loc_\varphi + Y^\exp} + I_{Y^\exp} \xrightarrow{\cong} I_{X^\loc_\varphi + Y^\exp} + I_{Z^\in}
\end{array}
$$

As a composition of bijections, $\omega$ is a bijection, and hence it is a morphism in $\mathbf{W}$.

It only remains to prove that composition of wiring diagrams is associative. Consider $V \xrightarrow{\Theta} X \xrightarrow{\Phi} Y \xrightarrow{\Psi} Z$ in $\mathbf{W}$, and let $\Lambda = (\Psi \circ \Phi) \circ \Theta, \Gamma = \Psi \circ (\Phi \circ \Theta) : V \to Z$ with corresponding typed bijections $\lambda$ and $\gamma$, respectively. We readily see that $\lambda^\out = \gamma^\out$ by the associativity of composition in $\mathbf{TFS}$. Proving that $\lambda^\in = \gamma^\in$ is equivalent to showing that the diagram below commutes:

$$\begin{align*}
V^\out + Z^\in & \xrightarrow{\nabla + I} V^\out + V^\out + Z^\in \\
I + \theta^\out + I & \xrightarrow{I + \psi^\out + I} V^\out + X^\out + Z^\in \\
V^\out + Y^\in & \xrightarrow{\nabla + I} V^\out + X^\out + Y^\out + Z^\in \\
I + \theta^\out + I & \xrightarrow{I + I + \phi^\out + I} V^\out + X^\out + Y^\out + Z^\in
\end{align*}
$$

Although the middle square in (6) does not commute by itself, the composite of the last two maps coequalizes it; that is, the two composite morphisms $V^\out + X^\out + Y^\in \to V^\out + Z^\in$ agree. This follows formally from the fact that
+ is a coproduct, using standard facts about composing coproducts and codiagonals, or it can be shown concretely using elements and case analysis. For completeness, we include a sketch of the formal argument.

\[
(\nabla + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1})(\mathbb{1} + \phi^\text{out} + \mathbb{1})(\mathbb{1} + \psi^\text{in} + \mathbb{1})(\nabla + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1}) = \\
(\nabla + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1})(\mathbb{1} + \phi^\text{out} + \mathbb{1})(\nabla + \mathbb{1} + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \psi^\text{in}) = \\
(\nabla + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1})(\mathbb{1} + \phi^\text{out} + \mathbb{1})(\nabla + \mathbb{1} + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1} + \mathbb{1})(\mathbb{1} + \mathbb{1} + \phi^\text{out} + \mathbb{1})(\mathbb{1} + \mathbb{1} + \psi^\text{in}) = \\
(\nabla + \mathbb{1})(\mathbb{1} + \theta^\text{out} + \mathbb{1})(\mathbb{1} + \nabla + \mathbb{1})(\mathbb{1} + \mathbb{1} + \phi^\text{out} + \mathbb{1})(\mathbb{1} + \mathbb{1} + \psi^\text{in}).
\]

\[\square\]

The following remark explains that our pictures of wiring diagrams are not completely ad hoc—they are depictions of 1-dimensional oriented manifolds with boundary. The boxes in our diagrams simply tie together the positively- and negatively-oriented components of an individual oriented 0-manifold.

**Remark 3.13.** Let \(1\text{-Cob}\) denote the symmetric monoidal category of oriented 0-manifolds and 1-dimensional cobordisms between them. Let \(W^*\) denote the category of wiring diagrams over the terminal SMC. Then there is a faithful, essentially surjective, strong monoidal functor

\[W^* \rightarrow 1\text{-Cob},\]

sending a box \((X^\text{in}, X^\text{out})\) to the oriented 0-manifold \(X^\text{in} + X^\text{out}\) where \(X^\text{in}\) is oriented positively and \(X^\text{out}\) negatively. Under this functor, a wiring diagram \(\Phi: X \rightarrow Y\) is sent to a 1-dimensional cobordism that has no closed loops. A connected component of such a cobordism can be identified with either its left or right endpoint, which correspond to the domain or codomain of the bijection \(\varphi: X^\text{in} + Y^\text{out} \rightarrow X^\text{out} + Y^\text{in}\).

The no passing wires condition on morphisms (cobordisms) \(X \rightarrow Y\) (see Definition 3.5) can be interpreted as saying that the induced map on components, from those of the codomain \(Y\) to those of the cobordism itself, is injective. This condition assures that no new closed loops are formed under composition of these cobordisms.

Note, however, that the 0-dimensional manifolds and cobordisms we are discussing in this remark are about the connection patterns between boxes, but have no relation to the manifolds we use to type ports in Definition 2.4.

Let \(\Phi: X \rightarrow Y\) be a wiring diagram in \(W\) with a corresponding typed bijection \(\varphi: X^\text{in} + Y^\text{out} \rightarrow X^\text{out} + Y^\text{in}\). Applying the dependent product functor (see Definition 2.5), we obtain a diffeomorphism of manifolds

\[\overline{\varphi}: X^\text{out} \times Y^\text{in} \rightarrow X^\text{in} \times Y^\text{out}.\]

By the no passing wires condition (and reasoning as in Remark 3.7), \(\overline{\varphi}\) has component maps

\[
\left\{
\begin{align*}
\overline{\varphi}^\text{in}: & X^\text{out} \times Y^\text{in} \rightarrow X^\text{in} \\
\overline{\varphi}^\text{out}: & X^\text{out} \rightarrow Y^\text{out}
\end{align*}
\right.
\]

We may also apply the dependent product functor to the commutative diagrams in (5), which define wiring diagram composition. Note that the image of the
codiagonal $\nabla: X_{\text{out}} + X_{\text{out}} \rightarrow X_{\text{out}}$ under the dependent product is the diagonal map $\Delta: X_{\text{out}} \rightarrow X_{\text{out}} \times X_{\text{out}}$. Thus we have the following commutative diagrams:

\[
\begin{array}{c}
\xymatrix{
X_{\text{out}} \times Z_{\text{in}} \ar[r]^{\omega_{\text{in}}} \ar[d]_{\Delta \times 1} & X_{\text{in}} \ar[d]_{\omega_{\text{out}}} \\
X_{\text{out}} \times X_{\text{out}} \times Z_{\text{in}} \ar[r]_{1 \times \psi_{\text{out}} \times 1} & X_{\text{out}} \times Y_{\text{out}} \times Z_{\text{in}} 
}
\end{array}
\]

4. The Algebra of Open Systems

In this section we define an algebra $G: (W, \oplus, 0) \rightarrow (\text{Set}, \times, \star)$ (see Definition 2.3) of general open dynamical systems. A $W$-algebra can be thought of as a choice of semantics for the syntax of wiring diagrams—a set of possible meanings for boxes and wiring diagrams. As in Definition 2.1, we may use this to construct the corresponding operad algebra $OG: OW \rightarrow \text{Sets}$. We will first define how $G$ acts on boxes, and then on monoidal products and wiring diagrams, finally proving it is a $W$-algebra in Proposition 4.6. We now revisit Example 1.1 for inspiration.

Example 4.1. As the textbook exercise [BD65, Problem 7.21] prompts, let’s begin by writing down the system of equations that governs the amount of salt $Q_i$ within the tanks $X_i$. This can be done by using dimensional analysis for each port of $X_i$ to find the rate of salt being carried in ounces per minute, and then equating the rate $\dot{Q}_i$ to the sum across these rates for $X_{i\text{in}}$ ports minus $X_{i\text{out}}$ ports.

\[
\dot{Q}_1 = \frac{Q_1 \text{oz}}{30 \text{gal}} \cdot \frac{3 \text{gal}}{\text{min}} + Q_2 \text{oz} \cdot \frac{1.5 \text{gal}}{\text{min}} + \frac{1 \text{oz}}{1 \text{gal}} \cdot \frac{1.5 \text{gal}}{\text{min}}
\]

\[
\dot{Q}_2 = Q_2 \text{oz} \cdot \frac{(1.5 + 2.5) \text{gal}}{20 \text{gal}} \cdot \frac{3 \text{gal}}{30 \text{gal}} + Q_1 \text{oz} \cdot \frac{3 \text{oz}}{30 \text{gal}} \cdot \frac{1 \text{gal}}{\text{min}}
\]

Dropping the physical units, we are left with the following system of ODEs:

\[
\begin{cases}
\dot{Q}_1 = -0.1Q_1 + 0.075Q_2 + 1.5 \\
\dot{Q}_2 = 0.1Q_1 - 0.2Q_2 + 3
\end{cases}
\]

The equations in (8) include a hidden step in which the connection pattern in Figure 1 is used. The purpose of our work is to explain this step and make it explicit. Each box in a wiring diagram should only “know” about its own inputs and outputs, and not how they are connected to others. That is, we can only define an element of $G(X_i)$ by expressing $\dot{Q}_i$ only in terms of $Q_i$ and $X_{i\text{in}}$. In Example 5.3, we will explicitly compute the element that encodes the system in Example 1.1. From there we can recover (8) by using the wiring diagram $\Phi: X_1, X_2 \rightarrow Y$ with Remark 3.6 to establish that $X_{1\text{in}} = X_{2\text{out}}$ and $X_{2\text{in}} = X_{1\text{out}}$. We then use the readout maps (see Definition 2.9) $X_{2\text{in}} = 0.75Q_2$ and $X_{1\text{out}} = 0.7Q_1$ to write the system without wires, as in (8). The necessary data can all be neatly packaged via the open system notion established in Definition 2.9.
Definition 4.2. Let $X \in \text{Ob } \mathbf{W}$. The set of open systems on $X$, denoted $\mathcal{G}(X)$, is defined as

$$\mathcal{G}(X) = \{(S, f) \mid S \in \text{Ob } \mathbf{TFS}, (\overline{S}, X^{\text{in}}, X^{\text{out}}, f) \in \text{Ob } \mathbf{ODS}\}.$$ 

We call $S$ the set of state variables and its dependent product $\overline{S}$ the state space.

Recall from Remark 2.7 that $\text{Man}$ is small, so the collection $\mathcal{G}(X)$ of open systems on $X$ is indeed a set.

Remark 4.3. One may also encode an initial condition in $\mathcal{G}$ by using $\text{Man}_*$ instead of $\text{Man}$ in Remark 2.7 as the default choice of cartesian category, where $\text{Man}_*$ is the category of pointed smooth manifolds and base point preserving smooth maps. The base point represents the initialization of the state variables.

In the following definition, one may note an interesting resemblance between its diagrams (9) and those in (5).

Definition 4.4. Let $\Phi: X \to Y$ be in $\mathbf{W}$. Then $\mathcal{G}(\Phi): \mathcal{G}(X) \to \mathcal{G}(Y)$ is given by $(S, f) \mapsto (\mathcal{G}(\Phi) S, \mathcal{G}(\Phi) f)$, where $\mathcal{G}(\Phi) S = S$ and $g = \mathcal{G}(\Phi) f : \overline{S} \times Y^{\text{in}} \to T\overline{S} \times Y^{\text{out}}$ is defined as the dashed arrows $(g^{\text{in}}, g^{\text{out}})$ that make the diagrams below commute:

\[
\begin{array}{ccc}
\overline{S} \times Y^{\text{in}} & \xrightarrow{g^{\text{in}}} & TS \\
\downarrow \Delta \times 1_{Y^{\text{in}}} & & \uparrow f^{\text{in}} \\
\overline{S} \times S \times Y^{\text{in}} & \xrightarrow{f^{\text{out}}} & X^{\text{out}} \times Y^{\text{out}} \\
\downarrow 1_{\overline{S}} \times f^{\text{out}} \times 1_{Y^{\text{in}}} & & \downarrow 1_{X^{\text{out}}} \times \varphi^{\text{out}} \\
\overline{S} \times X^{\text{out}} \times Y^{\text{in}} & \xrightarrow{1_{\overline{S}} \times \varphi^{\text{out}}} & \overline{S} \times X^{\text{in}}
\end{array}
\]

Since $\mathcal{G}$ is a lax monoidal functor, it must be equipped with a coherence map that encodes its monoidal structure.

Definition 4.5. Let $X, X' \in \mathbf{W}$. Then $\mu : \mathcal{G}(X) \times \mathcal{G}(X') \to \mathcal{G}(X \oplus X')$ is given by $((S, f), (S', f')) \mapsto (S + S', f \times f')$, where $f \times f'$ is as in Lemma 2.13.

In contrast to the trivial equality $\mathcal{G}(\Phi) S = S$ found in Definition 4.4, in the operad setting we have $\mathcal{O} \mathcal{G}(\Phi)(S_1, \ldots, S_n) = \Pi_{i=1}^n S_i$. This follows by Definition 4.5. Thus the state variables of the larger box $Y$ are the sum of the state variables of its constituent boxes $X_i$.

As established in Definition 2.1, the coherence map $\mu$ allows us to define the operad algebra $\mathcal{O} \mathcal{G}$ from $\mathcal{G}$. Next we define how $\mathcal{G}$ acts on wiring diagrams.

Proposition 4.6. Let $X \xrightarrow{\Psi} Y \xrightarrow{\Phi} Z$ be composable morphisms in $\mathbf{W}$. Then $\mathcal{G}(\Psi \circ \Phi) = \mathcal{G}(\Phi) \circ \mathcal{G}(\Psi)$. Therefore, $\mathcal{G} : \mathbf{W} \to \text{Set}$ is a functor, and together with the coherence map $\mu$, forms a $\mathbf{W}$-algebra.

Proof. Immediately we have $\mathcal{G}(\Psi \circ \Phi) S = S = \mathcal{G}(\Psi)(\mathcal{G}(\Phi) S)$. Thus if we let $h = \mathcal{G}(\Psi \circ \Phi) f$ and $k = \mathcal{G}(\Phi)(\mathcal{G}(\Psi) f)$, it suffices to show $h = k$. One readily sees that $h^{\text{out}} = k^{\text{out}}$. We use (7) and (9) to produce the following diagram; proving
it commutes is equivalent to proving that that \( h^{\text{in}} = k^{\text{in}} \).

\[
\begin{array}{c}
\mathcal{S} \times \overline{Z}^{\text{in}} \\
\downarrow \Delta \times \mathbb{1} \\
\mathcal{S} \times \mathcal{S} \times \overline{Z}^{\text{in}} \\
\downarrow 1 \times f^{\text{out}} \times \mathbb{1} \\
\mathcal{S} \times \overline{Y}^{\text{in}} \\
\downarrow \Delta \times \mathbb{1} \\
\mathcal{S} \times \overline{Y}^{\text{in}} \\
\downarrow 1 \times f^{\text{out}} \times \mathbb{1} \\
\mathcal{S} \times \overline{X}^{\text{in}} \\
\downarrow f^{\text{in}} \\
\mathcal{T} \mathcal{S}
\end{array}
\]

This commutative diagram is in some sense dual to the one for associativity in Proposition 3.12 (6). Although the middle square fails to commute by itself, the composite of the first two maps equalizes it; that is, the two composite morphisms \( \mathcal{S} \times \overline{Z}^{\text{in}} \to \mathcal{S} \times \overline{X}^{\text{out}} \times \overline{Y}^{\text{in}} \) agree. This follows formally from the fact that \( \times \) is a product, using standard facts about diagonals, or it can be shown concretely by considering elements.

Since we showed the analogous result formally in the proof of Proposition 3.12, we show it concretely using elements this time. Let \( (s, z) \in \mathcal{S} \times \overline{Z}^{\text{in}} \) be an arbitrary element. Composing six morphisms \( \mathcal{S} \times \overline{Z}^{\text{in}} \to \mathcal{S} \times \overline{X}^{\text{out}} \times \overline{Y}^{\text{in}} \) through the left of the diagram gives the same answer as composing through the right; namely,

\[
(s, f^{\text{out}}(s), \psi^{\text{in}}(f^{\text{out}}(s), z)) \in \mathcal{S} \times \overline{X}^{\text{out}} \times \overline{Y}^{\text{in}}.
\]

Since the diagram commutes, we have shown that the pair \((\mathcal{G}, \mu)\) constitutes a lax monoidal functor \( W \to \text{Set} \), i.e. a \( W \)-algebra. \(\square\)

5. The Subalgebra of Linear Open Systems

In this section, we define an algebra \( \mathcal{L} : W_{\text{Lin}} \to \text{Set} \), which encodes linear open systems. Here \( W_{\text{Lin}} \) is the category of \text{Lin}-typed wiring diagrams, as in Remark 3.2. Of course, one can use Definition 2.1 to construct an operad algebra \( \mathcal{O} \mathcal{L} : \mathcal{O}W_{\text{Lin}} \to \text{Sets} \).

**Definition 5.1.** Let \( X \in \text{Ob} W_{\text{Lin}} \). Then the set of linear open systems on \( X \), denoted \( \mathcal{L} \), is defined as

\[
\mathcal{L}(X) := \{ (S, f) \mid S \in \text{Ob} TFS_{\text{Lin}}, (\mathcal{S}, \overline{X}^{\text{in}}, \overline{X}^{\text{out}}, f) \in \text{Ob} ODS_{\text{Lin}} \}.
\]
Remark 5.2. The coherence map \( \mu_{\text{Lin}} : \mathcal{L}(X) \times \mathcal{L}(X) \to \mathcal{L}(X \oplus X') \) is given, as in Definition 4.5, by \( ((S, f), (S', f')) \to (S + S', f \times f') \).

Example 5.3. As promised in Example 4.1, we can write the open systems for \( X_i \) in Example 1.1 as elements of \( \mathcal{L}(X_i) \). The linear open systems below in (11) represent \( f_1 \) and \( f_2 \), respectively.

\[
(11) \begin{bmatrix} \dot{Q}_1 \\ X_{1a}^{\text{out}} \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ X_{1a}^{\text{in}} \\ X_{1b}^{\text{in}} \end{bmatrix}, \quad \begin{bmatrix} \dot{Q}_2 \\ X_{2a}^{\text{out}} \\ X_{2b}^{\text{out}} \end{bmatrix} = \begin{bmatrix} -2 & 1 & 1 \\ .125 & 0 & 0 \end{bmatrix} \begin{bmatrix} Q_2 \\ X_{2a}^{\text{in}} \\ X_{2b}^{\text{in}} \end{bmatrix}
\]

As a sanity check, we recover the equation from (8):

\[
\begin{align*}
\dot{Q}_1 &= -1Q_1 + X_{1a}^{\text{in}} + X_{1b}^{\text{in}} = -1Q_1 + 1.5 + X_{2b}^{\text{out}} = -1Q_1 + .075Q_2 + 1.5 \\
\dot{Q}_2 &= -2Q_2 + X_{2b}^{\text{in}} + X_{2b}^{\text{in}} = -2Q_2 + 3 + X_{1a}^{\text{out}} = -2Q_2 + .1Q_1 + 3
\end{align*}
\]

Note the proportion of zeros and ones in the \( f \)-matrices of (11)—this is perhaps why the making explicit of these details was an afterthought in (8). Because we may have arbitrary nonconstant coefficients, our formalism can capture more intricate systems.

We will show how \( \mathcal{L} \) acts on wiring diagrams in Definition 5.7 by using a new, simpler decomposition of wiring diagrams. We note that \( f \) is now a morphism in \( \text{Lin} \), which enjoys special properties—in particular it is an additive category, as seen by the fact that there is an equivalence of categories \( \text{Lin} \cong \text{Vect}_\mathbb{R} \). Specifically, finite products and finite coproducts are isomorphic. Hence a linear map \( f : A_1 \times A_2 \to B_1 \times B_2 \) decomposes universally into four linear maps \( f^{i,j} : A_i \to B_j \) where \( i, j \in \{1, 2\} \), which together are naturally equivalent to the whole map by various universal properties. To be more concrete, as we shall see in (12), these four linear maps are literally four quadrants of the matrix that represents \( f \).

Example 5.4. Let’s return to Example 1.1 and use Remark 5.2 to define the combined tank system

\[
(Q, f) = \mu_{\text{Lin}}((\{Q_1\}, f_1), (\{Q_2\}, f_2)) = (\{Q_1, Q_2\}, f_1 \times f_2),
\]

where \( f : \overline{Q} \times \overline{X}^{\text{in}} \to \overline{TQ} \times \overline{X}^{\text{out}} \) is a linear transformation that decomposes into the four components

\[
\begin{align*}
\begin{bmatrix} f^{Q, Q} \\ f^{Q, X} \end{bmatrix} : \overline{Q} \to \overline{TQ} & \quad \begin{bmatrix} f^{Q, X} \end{bmatrix} : \overline{X}^{\text{in}} \to \overline{TQ} \\
\begin{bmatrix} f^{X, Q} \\ f^{X, X} \end{bmatrix} : \overline{X}^{\text{out}} \to \overline{X}^{\text{out}}
\end{align*}
\]

We may then write down the system for Example 1.1 in terms of these components:

\[
(12) \begin{bmatrix} \hat{Q} \\ X^{\text{out}} \end{bmatrix} = \begin{bmatrix} f^{Q, Q} & f^{Q, X} \\ f^{X, Q} & f^{X, X} \end{bmatrix} \begin{bmatrix} Q \\ X^{\text{in}} \end{bmatrix}
\]

We will exploit this form in Definition 5.7 to make simpler definitions and computations, by encoding \( g = \mathcal{L}(\Phi)f \) into one matrix equation. To do so we will first need to encode \( \Phi \) in \( W_{\text{Lin}} \) as a matrix. Since \( \overline{\Phi} : \overline{X}^{\text{out}} \times \overline{Y}^{\text{in}} \to \overline{X}^{\text{in}} \times \overline{Y}^{\text{out}} \) is a linear transformation, it is naturally realizable as a matrix.

One can think of \( \overline{\Phi} \) as a permutation matrix that can be encoded as a block matrix consisting of identity and zero matrix blocks. An identity matrix in block
entry \((i, j)\) represents the fact that the port whose state space corresponds to row \(i\) and the one whose state space corresponds to column \(j\) get linked by \(\Phi\).

**Example 5.5.** We encode the bijection \(\varphi\) from Table 1 as a matrix \(\overrightarrow{\varphi}\) below:

\[
\begin{bmatrix}
X_{\text{out}}^{1a} \\
X_{\text{out}}^{2a} \\
Y_{\text{in}}^{a} \\
Y_{\text{out}}^{b}
\end{bmatrix}
= 
\begin{bmatrix}
0 & 0 & I & 0 & 0 \\
0 & 0 & 0 & 0 & I \\
I & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & I & 0
\end{bmatrix}
\begin{bmatrix}
X_{\text{in}}^{1a} \\
X_{\text{in}}^{1b} \\
X_{\text{in}}^{2a} \\
X_{\text{in}}^{2b} \\
Y_{\text{out}}^{a}
\end{bmatrix}
\]

Recalling Example 3.4, all of these ports are typed in \(\mathbb{R}\), so we have \(I = 1\) in (13). In general, the dimension of each \(I\) is equal to the dimension of the corresponding state space. The formula in (13) is true independent of the typing.

As we did for \(f\) in (12), we may decompose \(\overrightarrow{\varphi}\) into matrix blocks:

\[
\overrightarrow{\varphi} = \begin{bmatrix}
\varphi_{X,X} & \varphi_{X,Y} \\
\varphi_{Y,X} & \varphi_{Y,Y}
\end{bmatrix}
\]

where the four maps of our decomposition are

\[
\begin{align*}
\varphi_{X,X} &: \ X_{\text{out}} \to X_{\text{in}} \\
\varphi_{Y,X} &: \ Y_{\text{in}} \to X_{\text{out}} \\
\varphi_{X,Y} &: \ X_{\text{out}} \to Y_{\text{out}} \\
\varphi_{Y,Y} &: \ Y_{\text{in}} \to Y_{\text{out}}
\end{align*}
\]

**Remark 5.6.** By virtue of the no passing wires condition in Definition 3.5, the \(\varphi_{Y,Y}\) block of a wiring diagram \(\Phi: X \to Y\) must be the zero matrix. In addition, our bijectivity condition implies that \(\varphi\) has precisely one nonzero entry in each row and column.

**Definition 5.7.** Let \(\Phi: X \to Y\) be in \(\mathbf{W}_{\mathbf{Lin}}\). Then, as in Definition 4.4, we define \(\mathcal{L}(\Phi)(S, f) := (S, g)\), where \(g\) is defined below:

\[
g = \begin{bmatrix}
g^{S,S} & g^{S,X} \\
g^{X,S} & g^{X,X}
\end{bmatrix} = \begin{bmatrix}
f^{S,X} & 0 \\
0 & I
\end{bmatrix} \overrightarrow{\varphi} \begin{bmatrix}
f^{X,S} & 0 \\
0 & I
\end{bmatrix} + \begin{bmatrix}
f^{S,S} & 0 \\
0 & 0
\end{bmatrix}
\]

\[
(14)
\]

This is really just a linear version of the commutative diagrams in (9). For example, the equation \(g^{S,S} = f^{S,X} \varphi_{X,X} f^{X,S} + f^{S,S}\) can be read off the diagram for \(g^{\text{in}}\) in (9), using the additivity of \(\mathbf{Lin}\).

**Example 5.8.** We can now finish our work with Example 1.1 by writing down the open system \(g = \mathcal{L}(\Phi) f \in \mathcal{L}(Y)\) describing the outer box \(Y\) that encodes our entire open system, \(g: \dot{Q} \times Y_{\text{in}} \to T\dot{Q} \times Y_{\text{out}}\).

\[
\begin{bmatrix}
\dot{Q}_1 \\
\dot{Q}_2 \\
Y_{\text{out}}
\end{bmatrix} = 
\begin{bmatrix}
-.1 & .075 & 0 & 1 \\
.1 & -.2 & 1 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
Q_1 \\
Q_2 \\
Y_{\text{in}}^{a} \\
Y_{\text{in}}^{b}
\end{bmatrix}
\]
We will prove that $L$ is an algebra, by first expressing $\omega$, the matrix corresponding to the composed wiring diagram $\Omega = \Psi \circ \Phi$, using a matrix equation in terms of $\varphi$ and $\psi$. To do so, we simply recast (5) in matrix form below.

$$\omega = \begin{bmatrix}
\omega_{X,Y} & 0 \\
0 & I \\
\omega_{Z,X} & 0 \\
0 & 0
\end{bmatrix} = \begin{bmatrix}
\psi Y,X & \varphi X,Y \\
\varphi X,X & \psi Y,Z \\
\psi Z,X & \varphi X,Y \\
0 & 0
\end{bmatrix}$$

Proposition 5.9. Let $X \xrightarrow{\Phi} Y \xrightarrow{\Psi} Z$ be composable morphisms in $W_{\text{Lin}}$. Then $L(\Psi \circ \Phi) = L(\Psi) \circ L(\Phi)$. Therefore $L$, together with $\mu_{\text{Lin}}$, is a $W_{\text{Lin}}$-algebra.

Proof. We immediately have $L(\Psi \circ \Phi)S = L(\Psi)(L(\Phi)S)$. Let $h := L(\Psi \circ \Phi)f$ and $k := L(\Psi)(L(\Phi)f)$. We must show $h = k$.

Let $g = L(\Phi)f$ and $\Omega = \Psi \circ \Phi$ with corresponding matrix $\varpi$. It is then straightforward matrix arithmetic to see that

$$k = L(\Psi)g = \begin{bmatrix}
g_{S,Y} & 0 \\
0 & I \\
g_{S,S} & 0
\end{bmatrix} = \begin{bmatrix}
\psi Y,S & \varphi X,S \\
\varphi X,X & \psi Y,Z \\
0 & 0
\end{bmatrix}$$

Therefore, the pair $(L, \mu_{\text{Lin}})$ constitutes a lax monoidal functor $W_{\text{Lin}} \rightarrow \text{Set}$, i.e. a $W_{\text{Lin}}$-algebra.

5.1. The relationship between $G$ and $L$. We want to compare the $W_{\text{Lin}}$-algebra $L$, defined above, to the $W$-algebra $G$, defined in Section 4. Because they have different sources, $L$ is technically not a subalgebra of $G$, although it is close to being one in the sense of the following diagram.

$$\begin{array}{ccc}
W_{\text{Lin}} & \xrightarrow{W_i} & W \\
\downarrow{\epsilon} & \searrow{\varphi} \\
\mathcal{L} & \searrow{G} & \text{Set}
\end{array}$$

Here, the natural inclusion $W_i : W_{\text{Lin}} \rightarrow W$ corresponds to $i : \text{Lin} \rightarrow \text{Man}$, and we have a natural transformation $\epsilon : \mathcal{L} \rightarrow G \circ i$. Hence for each $X \in \text{Ob} W_{\text{Lin}}$, we have a function $\epsilon_X : L(X) \rightarrow G(i(X)) = G(X)$ that sends the linear open system $(S, f) \in L(X)$ to the open system $(\text{TFS}_i(S, i(f)) = (S, f) \in G(X)$.

REFERENCES

[AMMO10] Rob Arthan, Ursula Martin, Erik A. Mathiesen, and Paulo Oliva. A general framework for sound and complete Floyd-Hoare logics. ACM Trans. Comput. Log., 11(1):Art. 7, 31, 2010.

[Awo10] Steve Awodey. Category theory, volume 52 of Oxford Logic Guides. Oxford University Press, Oxford, second edition, 2010.
[BB12] John C. Baez and Jacob Biamonte. A course on quantum techniques for stochastic mechanics. ePrint online: www.arXiv.org/abs/1209.3632, 2012.

[BD65] William E. Boyce and Richard C. DiPrima. Elementary differential equations and boundary value problems. John Wiley & Sons, Inc., New York-London-Sydney, 1965.

[BS11] J. Baez and M. Stay. Physics, topology, logic and computation: a Rosetta Stone. In New structures for physics, volume 813 of Lecture Notes in Phys., pages 95–172. Springer, Heidelberg, 2011.

[Coe13] Bob Coecke. An alternative gospel of structure: order, composition, processes. ePrint online: www.arXiv.org/abs/1307.4038, 2013.

[DL10] Lee DeVille and Eugene Lerman. Dynamics on networks i. combinatorial categories of modular continuous-time systems. ePrint online: www.arXiv.org/abs/1008.5359, 2010.

[DL12] Lee DeVille and Eugene Lerman. Dynamics on networks of manifolds. ePrint online: www.arXiv.org/abs/1208.1513, 2012.

[DL14] Lee DeVille and Eugene Lerman. Modular dynamical systems on networks. JEMS (to appear). ePrint online: www.arXiv.org/abs/1303.3907, 2014.

[JS91] Andr´e Joyal and Ross Street. The geometry of tensor calculus. I. Adv. Math., 88(1):55–112, 1991.

[JSV96] Andr´e Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. Math. Proc. Cambridge Philos. Soc., 119(3):447–468, 1996.

[KFA69] R. E. Kalman, P. L. Falb, and M. A. Arbib. Topics in mathematical system theory. McGraw-Hill Book Co., New York-Toronto, Ont.-London, 1969.

[Koe67] Arthur Koestler. The Ghost in the Machine. Hutchinson & Co., 1967.

[Lei04] Tom Leinster. Higher operads, higher categories, volume 298 of London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 2004.

[ML98] Saunders Mac Lane. Categories for the working mathematician, volume 5 of Graduate Texts in Mathematics. Springer-Verlag, New York, second edition, 1998.

[RS13] Dylan Rupel and David I. Spivak. The operad of temporal wiring diagrams: formalizing a graphical language for discrete-time processes. ePrint online: www.arXiv.org/abs/1307.6894, 2013.

[RS15] Dylan Rupel and David I. Spivak. 0-dimensional tqfts and traced monoidal categories. (In preparation), 2015.

[Sco71] Dana Scott. The lattice of flow diagrams. In Symposium on Semantics of Algorithmic Languages, pages 311–366. Lecture Notes in Mathematics, Vol. 188. Springer, Berlin, 1971.

[Spi65] Michael Spivak. Calculus on manifolds. A modern approach to classical theorems of advanced calculus. W. A. Benjamin, Inc., New York-Amsterdam, 1965.

[spi13] David I. Spivak. The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits. ePrint online: www.arXiv.org/abs/arXiv:1305.0297, 2013.

[spi14] David I. Spivak. Category theory for the sciences. MIT Press, 2014.

[Tes12] Gerald Teschl. Ordinary differential equations and dynamical systems, volume 140 of Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2012.

[War83] Frank W. Warner. Foundations of differentiable manifolds and Lie groups, volume 94 of Graduate Texts in Mathematics. Springer-Verlag, New York-Berlin, 1983. Corrected reprint of the 1971 edition.