THE OPERAD OF WIRING DIAGRAMS: FORMALIZING A GRAPHICAL LANGUAGE FOR DATABASES, RECURSION, AND PLUG-AND-PLAY CIRCUITS

DAVID I. SPIVAK

ABSTRACT. Wiring diagrams, as seen in digital circuits, can be nested hierarchically and thus have an aspect of self-similarity. We show that wiring diagrams form the morphisms of an operad $\mathcal{T}$, capturing this self-similarity. We discuss the algebra $\mathcal{R}$el of mathematical relations on $\mathcal{T}$, and in so doing use wiring diagrams as a graphical language with which to structure queries on relational databases. We give the example of circuit diagrams as a special case. We move on to show how plug-and-play devices and also recursion can be formulated in the operadic framework as well. Throughout we include many examples and figures.

CONTENTS

1. INTRODUCTION

In the classic book *The Society of Mind* [Min], Marvin Minsky proposes that the human mind is composed of a society of agencies, each of which is composed of a society of smaller agencies, and so on. While the lower-level agencies could not be considered intelligent, their composition into structured wholes yields behavior that is markedly intelligent. He argues that this self-similar or recursive structure emerges naturally and is efficient for processing information. In this paper we offer a mathematical framework for modeling self-similar information processing structures of a certain kind.

In particular we will be interested in the self-similar nature of wiring diagrams, like those found in the design of digital circuits. One can wire together simple logic gates to form more complex ones, and wire these together to form still more complex circuits. Similarly, one may imagine that neurons can be wired together to form more complex information-processing centers in the brain, and that these can be wired together to form still more complex agencies.

We will discuss the so-called *wiring diagrams operad* $\mathcal{S}$, which models the notion of wiring smaller entities together to form larger ones. We will be careful to introduce
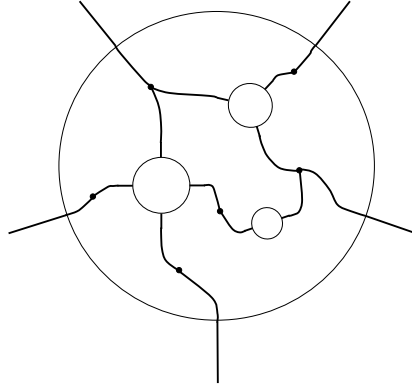
FIGURE 1.   An entity composed of three smaller entities, all wired together.

all the necessary operadic machinery, but it can be a bit dense for those who are unacquainted. For this reason, we will also draw plenty of pictures (such pictures constitute a large portion of the length of this document), which look like Figure 1.

Operads were invented in the 1970s by [BV] and [May]. We will recall the definition of operads in Definition 2.1.2, but it might be phrased in the current language using the slogan "a wiring diagram of wiring diagrams is a wiring diagram". That is, if each little circle in Figure 1 is filled in with its own little wiring diagram, the result will be a big outer circle with tiny ones wired together inside. This will be spelled out in Example 2.1.7.

Deciding what kind of object each circle represents (whether it be a digital circuit, an agency in the brain, a mathematical relation, etc.), and deciding how a wiring diagram such as that in Figure 1 produces a new such object from its constituent objects, is tantamount to choosing an *algebra* on the wiring diagrams operad. We will concentrate on the algebra $\mathcal{R}$el of mathematical relations, in which an outer relation is built by applying a wiring diagram to a set of constituent relations. This algebra $\mathcal{R}$el has immediate applications to visually representing relational database queries.

(2)
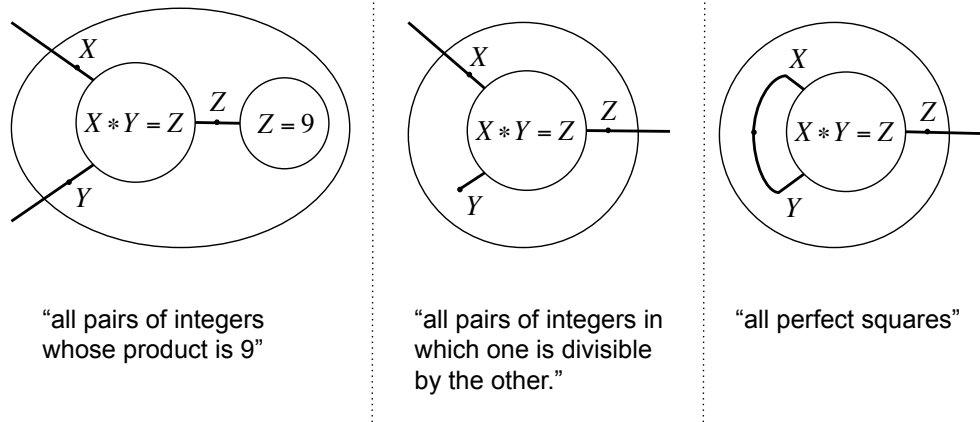
But as the title of the paper suggests, one can also express recursion and so-called "plug-and-play", or hot-plugging, in these terms. See Section 5.

Another reason for focusing on the algebra of relations is that digital circuits fall easily within this framework and provide a nice source of concrete examples. As a more abstract example, the algebra of relations can also serve as a crude model for agencies in the brain. We simply think of each agency as maintaining a relationship on its inputs. For example a person might be said to maintain a relationship between what she sees, hears, feels, and does; this can be represented as a 4-ary relation. Similarly, the visual cortex might be said to maintain relationships between the signals it receives from the eyes and from the rest of the brain.

In [JS], it is shown that in the near future, cyber-physical systems will likely continue to blur the line between humans and the machines on which they work. One may imagine that the same hierarchical structures we use to conceive the human mind can be extended beyond the individual to encompass interactions between multiple humans and machines. Therefore, it is useful to have a single mathematical model, such as the operadic model presented throughout this paper, in which all such levels and layers can be discussed in a unified way.

More broadly, we hope this paper will introduce operads to a more general audience. The wiring diagrams operad discussed here is but a single possibility among many that may prove useful. Whenever one wants to discuss a situation in which structures are built out of other structures and the functionality follows suit, operads and their algebras may be employable.

1.1. **Related work.** There are of course many books on digital circuits, e.g. [Eme], and all that use wiring diagrams are related to the present paper, as we explain in Example 2.2.11. Another related concept is that of the propagator network ([RS]), in which computation is performed by a "network of autonomous but interconnected machines". Clearly, the underlying idea behind our work and theirs is quite similar: machines made of networked machines. This paper can be seen as providing a mathematical underpinning to this idea in terms of operads and their algebras. For example, no such distinction seems to be alluded to in the work of [RS].

Finally, there is a similarity between this work and previous work by the author, including [Sp1], [Sp2], and [Sp3]. All of these provide different conceptions of databases. The one that appears closest to the usual relational model is the present paper, whereas the one that is most aligned with basic category theory is [Sp3]. The category-theoretic intersection between these viewpoints seems to be that databases are nicely modeled by sheaves in various guises.

1.2. **Layout of this paper.** This paper will be structured as follows. In Section 2, we will define most of the necessary operadic machinery and introduce our main operad of interest, the singly-typed wiring diagrams operad $\mathcal{S}$. We will also discuss our main algebras of interest, the relational $\mathcal{S}$-algebras $\mathcal{R}el_A$. Examples in this section include digital circuits and the three queries in (2). Section 2 constitutes the heart of the paper; the rest of the paper amounts to variations and explorations of its theme.

In Section 3 we discuss the notion of operad-theoretic invariants of the relational $\mathcal{S}$-algebra. After proving that certain attempts to find such invariants come up short, we conjecture that there are no nontrivial invariants (see Conjecture 3.1.6). In Section 4 we generalize the operad $\mathcal{S}$ to a typed version $\mathcal{T}$ in which each wire carries values of a specific type. We show how all the relational $\mathcal{S}$-algebras $\mathcal{R}el_A$ extend to a single $\mathcal{T}$-algebra $\mathcal{R}el$. In this section we more carefully discuss database queries (conjunctive and disjunctive), and compare to the wiring diagrams operad. In Section 5 we show that $\mathcal{S}$ and $\mathcal{T}$ are in fact closed operads, and we give applications to recursion (in

particular the factorial function) and the idea of plug-and-play devices. Finally, in Section 6 we discuss possibilities for future work.

1.3. **Acknowledgments.** I'd like to thank my former UROP student Gaurav Singh, who was the first to write down a recursive function in terms of wiring diagrams. His diagram for the factorial function has a prominant role in this paper: see (18). I'd also like to thank Dave Balaban, Al Jones, Emmy Murphy, Eswaran Subrahmanian, and especially Nat Stapleton for many useful and interesting conversations.

2. Operads and wiring diagrams

In this section we introduce operads, as well as our main operad of interest: the singly-typed wiring diagrams operad $\mathcal{S}$. We will also introduce algebras on an operad, as well as our main algebra of interest: the relational $\mathcal{S}$-algebra $\mathcal{R}$el.

We assume that the reader is familiar with basic category theory. Introductions to the subject include [Mac] and [Awo] for pure mathematicians, [BW] and [Pie] for computer scientists, and [Sp1] for a general scientific audience. For those unacquainted with category theory, we hope that the diagrams will get across the basic idea.

Notation 2.0.1. We denote the category of sets by **Set** and the full subcategory of finite sets by **Fin** $\subseteq$ **Set**.

If $n \in \mathbb{N}$ is a natural number, we write $\underline{n} \in$ **Set** to denote the set $\underline{n} := \{1, 2, \ldots, n\}$. We may also use symbols such as n to denote finite sets, e.g. $n \in \mathrm{Ob}(\mathbf{Fin})$. Given a finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, we denote its cardinality by $|n| \in \mathbb{N}$. In other words, the following composite is the identity on $\mathbb{N}$:

$$\mathbb{N} \xrightarrow{n \mapsto \underline{n}} \mathrm{Ob}(\mathbf{Fin}) \xrightarrow{n \mapsto |n|} \mathbb{N}.$$

For a set X, we let $\mathbb{P}(X) = \{R \subseteq X\}$ denote its power set. We denote a one-element set by $\{*\}$.

2.1. **Operads.**

Warning 2.1.1. Throughout this paper, we use the word *operad* to mean what is generally called a *symmetric colored operad* or a *symmetric multicategory*. This abbreviated nomenclature is not new, for example it is used in [Lur]. Hopefully no confusion will arise. For a full treatment of operads, multicategories, and how they fit into a larger mathematical context, see [Lei].

Definition 2.1.2. An *operad* $\mathcal{O}$ consists of some constituents (A. objects, B. morphisms, C. identities, D. compositions) conforming to some laws (1. identity law, 2. associativity law). The required constituents are as follows.

    A. There is a collection $\mathrm{Ob}(\mathcal{O})$, each element of which is called an *object* of $\mathcal{O}$.

    B. For each object $y \in \mathrm{Ob}(\mathcal{O})$, finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, and n-indexed set of objects $x \colon n \to \mathrm{Ob}(\mathcal{O})$, there is a set $\mathcal{O}_n(x; y) \in \mathrm{Ob}(\mathbf{Set})$. Its elements are called *morphisms from x to y* in $\mathcal{O}$.

    C. For every object $x \in \mathrm{Ob}(\mathcal{O})$, there is a specified morphism denoted $\mathrm{id}_x \in \mathcal{O}_1(x; x)$ called *the identity morphism on x*.

    D. Let $s \colon m \to n$ be a morphism in **Fin**. Let $z \in \mathrm{Ob}(\mathcal{O})$ be an object, let $y \colon n \to \mathrm{Ob}(\mathcal{O})$ be an n-indexed set of objects, and let $x \colon m \to \mathrm{Ob}(\mathcal{O})$ be an m-indexed set of objects. For each element $i \in n$, write $m_i := s^{-1}(i)$ for the pre-image of s under i, and write $x_i = x|_{m_i} \colon m_i \to \mathrm{Ob}(\mathcal{O})$ for the restriction of x to m_i. Then there is a function

(3) $$\circ \colon \mathcal{O}_n(y; z) \times \prod_{i \in n} \mathcal{O}_{m_i}(x_i; y(i)) \longrightarrow \mathcal{O}_m(x; z),$$

called *the composition formula.*

Given an n-indexed set of objects $x \colon n \to \mathrm{Ob}(\mathcal{O})$ and an object $y \in \mathrm{Ob}(\mathcal{O})$, we sometimes abuse notation and denote the set of morphisms from x to y by $\mathcal{O}(x_1, \ldots, x_n; y)$.[1] We may write $\mathrm{Hom}_\mathcal{O}(x_1, \ldots, x_n; y)$, in place of $\mathcal{O}(x_1, \ldots, x_n; y)$, when convenient. We can denote a morphism $\phi \in \mathcal{O}_n(x; y)$ by $\phi \colon x \to y$ or by $\phi \colon (x_1, \ldots, x_n) \to y$; we say that each x_i is a *domain object* of ϕ and that y is the *codomain object* of ϕ. We use infix notation for the composition formula, e.g. writing $\psi \circ (\phi_1, \ldots, \phi_n)$.

Here are the laws that govern the above constituents:

1. for every $x_1, \ldots, x_n, y \in \mathrm{Ob}(\mathcal{O})$ and every morphism $\phi \colon (x_1, \ldots, x_n) \to y$, we have
$$\phi \circ (\mathrm{id}_{x_1}, \ldots, \mathrm{id}_{x_n}) = \phi \qquad \text{and} \qquad \mathrm{id}_y \circ \phi = \phi;$$
and;

2. an appropriate associative law.[2]

Remark 2.1.3. In this remark we will discuss the abuse of notation in Definition 2.1.2 and how it relates to an action of a symmetric group on each morphism set in our definition of operad. We follow the notation of Definition 2.1.2, especially following the use of subscripts in the composition formula.

Suppose that $\mathcal{O}$ is an operad, $z \in \mathrm{Ob}(\mathcal{O})$ is an object, $y \colon n \to \mathrm{Ob}(\mathcal{O})$ is an n-indexed set of objects, and $\phi \colon y \to z$ is a morphism. If we linearly order n, enabling us to write $\phi \colon (y(1), \ldots, y(|n|)) \to z$, then changing the linear ordering amounts to finding an isomorphism of finite sets $\sigma \colon m \xrightarrow{\cong} n$, where $|m| = |n|$. Let $x = y \circ \sigma$ and for each $i \in n$, note that $m_i = \sigma^{-1}(\{i\}) = \{\sigma^{-1}(i)\}$, so $x_i = x|_{\sigma^{-1}(i)} = y(i)$. Taking $\mathrm{id}_{x_i} \in \mathcal{O}_{m_i}(x_i; y(i))$ for each $i \in n$, and using the identity law, we find that the composition formula induces a bijection $\mathcal{O}_n(y; z) \xrightarrow{\cong} \mathcal{O}_m(x; z)$, which we might denote by

$$\sigma \colon \mathcal{O}(y(1), y(2), \ldots, y(n); z) \cong \mathcal{O}\big(y(\sigma(1)), y(\sigma(2)), \ldots, y(\sigma(n)); z\big).$$

In other words, there is an induced group action of $\mathrm{Aut}(n)$ on $\mathcal{O}_n(x; z)$, where $\mathrm{Aut}(n)$ is the group of permutations of an n-element set.

Throughout this paper, we will permit ourselves to abuse notation and speak of morphisms $\phi \colon (x_1, x_2, \ldots, x_n) \to y$ for a natural number $n \in \mathbb{N}$, without mentioning

---

[1]There are three abuses of notation when writing $\mathcal{O}(x_1, \ldots, x_n; y)$, which we will fix one by one. First, it confuses the set $n \in \mathrm{Ob}(\mathbf{Fin})$ with its cardinality $|n| \in \mathbb{N}$. But rather than writing $\mathcal{O}(x_1, \ldots, x_{|n|}; y)$, it would be more consistent to write $\mathcal{O}(x(1), \ldots, x(|n|); y)$, because we have assigned subscripts another meaning in part D. But even this notation unfoundedly suggests that the set n has been endowed with a linear ordering, which it has not. This may be seen as a more serious abuse, but see Remark 2.1.3.

[2]Here is the associative law. Let $m \xrightarrow{s} n \xrightarrow{t} p$ be composable morphisms in $\mathbf{Fin}$. Let $z \in \mathrm{Ob}(\mathcal{O})$ be an object, let $y \colon p \to \mathrm{Ob}(\mathcal{O})$, $x \colon n \to \mathrm{Ob}(\mathcal{O})$, and $w \colon m \to \mathrm{Ob}(\mathcal{O})$ respectively be a p-indexed, n-indexed, and m-indexed set of objects. For each $i \in p$, write $n_i = t^{-1}(i)$ for the pre-image and $x_i \colon n_i \to \mathrm{Ob}(\mathcal{O})$ for the restriction. Similarly, for each $k \in n$ write $m_k = s^{-1}(k)$ and $w_k \colon m_k \to \mathrm{Ob}(\mathcal{O})$; for each $i \in p$, write $m_{i,-} = (t \circ s)^{-1}(i)$ and $w_{i,-} \colon m_{i,-} \to \mathrm{Ob}(\mathcal{O})$; for each $j \in n_i$, write $m_{i,j} := s^{-1}(j)$ and $w_{i,j} \colon m_{i,j} \to \mathrm{Ob}(\mathcal{O})$. Then the diagram below commutes:

$$\mathcal{O}_p(y; z) \times \prod_{i \in p} \mathcal{O}_{n_i}(x_i; y(i)) \times \prod_{i \in p, \ j \in n_i} \mathcal{O}_{m_{i,j}}(w_{i,j}; x_i(j))$$

$$\mathcal{O}_n(x; z) \times \prod_{k \in n} \mathcal{O}_{m_k}(w_k; x(k)) \qquad\qquad \mathcal{O}_p(y; z) \times \prod_{i \in p} \mathcal{O}_{m_{i,-}}(w_{i,-}; y(i))$$

$$\mathcal{O}_m(w; z)$$

the abuse inherent in choosing an order, so long as it is clear that permuting the order of indices would not change anything up to canonical isomorphism.

Example 2.1.4. We define the operad of sets, denoted **Sets**, as follows. We put $\mathrm{Ob}(\mathbf{Sets}) := \mathrm{Ob}(\mathbf{Set})$. Given a natural number $n \in \mathbb{N}$ and objects $X_1, \ldots, X_n, Y \in \mathrm{Ob}(\mathbf{Sets})$, we define

$$\mathbf{Sets}(X_1, X_2, \ldots, X_n; Y) := \mathrm{Hom}_{\mathbf{Set}}(X_1 \times X_2 \times \cdots \times X_n, Y).$$

For any $X \in \mathrm{Ob}(\mathbf{Sets})$ the identity morphism $\mathrm{id}_X \colon X \to X$ is the same identity as that in **Set**.

The composition formula is as follows. Suppose given a set $Z \in \mathrm{Ob}(\mathbf{Set})$, a natural number $n \in \mathbb{N}$, sets $Y_1, \ldots, Y_n \in \mathrm{Ob}(\mathbf{Set})$, and for each i natural numbers $m_1, \ldots, m_n \in \mathbb{N}$ and sets $X_{i,1}, \ldots, X_{i,m_i} \in \mathrm{Ob}(\mathbf{Set})$. Suppose furthermore that we have composable morphisms: a function $g \colon Y_1 \times \cdots \times Y_n \to Z$ and for each $i \in \underline{n}$ a function $f_i \colon X_{i_1} \times \ldots \times X_{i,m_i} \to Y_i$. We need a function $X_{1,1} \times X_{1,2} \times \cdots \times X_{n,m_n} \to Z$, which we take to be the composite

$$(X_{1,1} \times \cdots \times X_{1,m_1}) \times \cdots \times (X_{n,1} \times \cdots \times X_{n,m_n}) \xrightarrow{f_1 \times \cdots \times f_n} Y_1 \times \cdots \times Y_n \xrightarrow{g} Z.$$

It is not hard to see that this composition formula is associative.

Example 2.1.5. Let **S** be any category with finite products. Such categories include: the category **Fin** of finite sets, the category **PrO** of preorders, the category **JLat** of *join-semilattices*, the category **Type** of types in a programming language, the category **Cat** of small categories, etc. We then define an operad **Ss** with

$$\mathrm{Ob}(\mathbf{Ss}) = \mathrm{Ob}(\mathbf{S}) \qquad \text{and} \qquad \mathbf{Ss}_n(X_1, \ldots, X_n; Y) = \mathrm{Hom}_{\mathbf{S}}(X_1 \times \cdots \times X_n, Y).$$

The composition formula is similar to that discussed in Example 2.1.4.

Example 2.1.6. The *commutative operad* $\mathcal{E}$ has one object, say $\mathrm{Ob}(\mathcal{E}) = \{\blacktriangle\}$, and for each $n \in \mathbb{N}$ it has a single n-ary morphism, $\mathcal{E}_n(\blacktriangle, \ldots, \blacktriangle; \blacktriangle) = \{\mu_n\}$. In fact $\mathcal{E}$ can be viewed as a special case of Example 2.1.5 where $\mathbf{S} = \underline{1}$ is the terminal category; i.e. $\mathcal{E} \cong \underline{1}\mathbf{s}$. We will see in Lemma 2.2.4 that $\mathcal{E}$ is the terminal object in the category of operads. (Again, recall from Warning 2.1.1 that all of our operads are actually symmetric colored operads, and we are saying that $\mathcal{E}$ is the terminal object in that category.)

Example 2.1.7 (**Main Operad Example**). In this example we discuss the operad $\mathcal{S}$ of singly-typed wiring diagrams. The objects of $\mathcal{S}$ are simply finite sets, $\mathrm{Ob}(\mathcal{S}) = \mathrm{Ob}(\mathbf{Fin})$. However, we refer to each object as a *singly-typed star* and we draw an object, say $X := \{a, b, c, d, e\} \in \mathrm{Ob}(\mathcal{S})$, as follows:

(4)
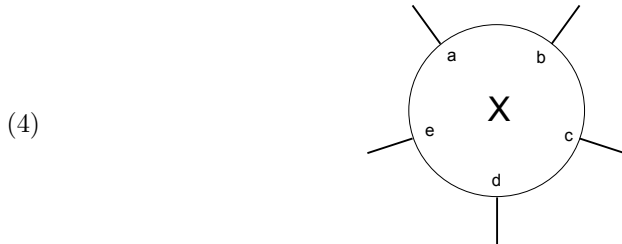
The order of the elements around the circle does not matter: X is just a set. Each element of X is called a *wire of X*.

Let $Y \in \mathrm{Ob}(\mathcal{S})$ be an object, let $n \in \mathbb{N}$ be a natural number, and let $X_1, \ldots, X_n \in \mathrm{Ob}(\mathcal{S})$ be singly-typed stars. A morphism $\phi \colon X_1, \ldots, X_n \to Y$ consists of a cospan, [3] i.e. a finite set $C \in \mathrm{Ob}(\mathbf{Fin})$ and functions

$$
\begin{array}{c}
Y \\
\downarrow {\scriptstyle g} \\
X_1 \amalg \cdots \amalg X_n \xrightarrow{\ f\ } C.
\end{array}
$$

We refer to such a morphism as an *singly-typed wiring diagram* (or a *wiring diagram* or even a *WD*, for short); we refer to the domain objects $X_1, \ldots, X_n$ as the *inner stars*, the codomain object Y as the *outer star*, and the set C as the *set of cables*. For any wire $y \in Y$ with $g(y) = c$ (respectively $x \in X_i$ with $f(x) = c$), we say that y (respectively x) is *soldered onto cable c*. Figure 6 should make all this quite a bit clearer.

Given composable morphisms $\phi' \colon Y_1, \ldots, Y_n \to Z$ and $\phi_1, \ldots, \phi_n$, where each $\phi_i \colon X_{i,1}, \ldots, X_{i,m_i} \to Y_i$, we can arrange their representing cospans in the form of the W-shaped diagram below. The composition formula is given by taking the pushout P of the middle square.

(7)
$$
\begin{array}{ccc}
 & & Z \\
 & & \downarrow {\scriptstyle g'} \\
\coprod_{i \in n} Y_i & \xrightarrow{\ f'\ } & D \\
{\scriptstyle \amalg_i g_i} \downarrow & & \downarrow \\
\coprod_{i \in n} \coprod_{j \in m_i} X_{i,j} \xrightarrow{\ \amalg_i f_i\ } & \coprod_{i \in n} C_i & \longrightarrow P
\end{array}
$$

Here $\phi' \circ (\phi_1, \ldots, \phi_n)$ is represented by the cospan $\coprod_{i \in n} \coprod_{j \in m_i} X_{i,j} \to P \leftarrow Z$.

The composition formula for $\phi' \circ (\phi_1, \ldots, \phi_n)$ is intuitive when drawing pictures. We begin with wiring diagrams ϕ_i of little inner stars inside intermediary stars and a wiring diagram ϕ' of intermediary stars inside a big outer star. The wires of the intermediary stars Y_i will match interior cables to exterior cables. The composition is visualized by first placing the little WDs ϕ_i into their respective outer stars Y_i, and then removing the intermediary circles Y_i, while preserving the connections they

---

[3] In fact, a morphism $\phi \colon (X_1, \ldots, X_n) \to Y$ in $\mathcal{S}$ is defined to be a cospan *up to isomorphism*. In other words, given a commutative diagram as below,

$$
\begin{array}{ccc}
 & C & \\
 {\scriptstyle f} \nearrow & \uparrow {\scriptstyle g} & \\
X_1 \amalg \cdots \amalg X_n & {\scriptstyle \cong} & Y \\
 {\scriptstyle f'} \searrow & \downarrow & \swarrow {\scriptstyle g'} \\
 & C' &
\end{array}
$$

we say that the cospans (f, g) and (f', g') *represent* the same morphism $X_1, \ldots, X_n \xrightarrow{\phi} Y$. This just means that two wiring diagrams are considered to be the same morphism in $\mathcal{S}$ if the only difference between them is a renaming of cables.
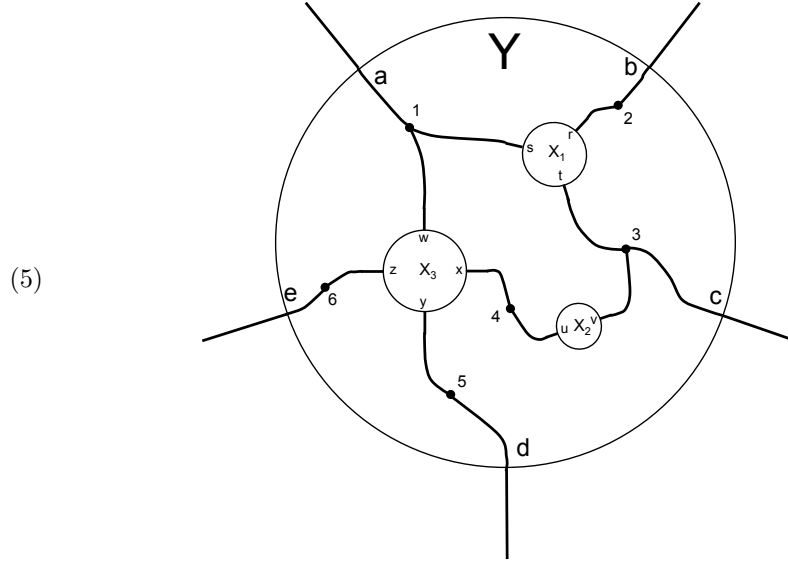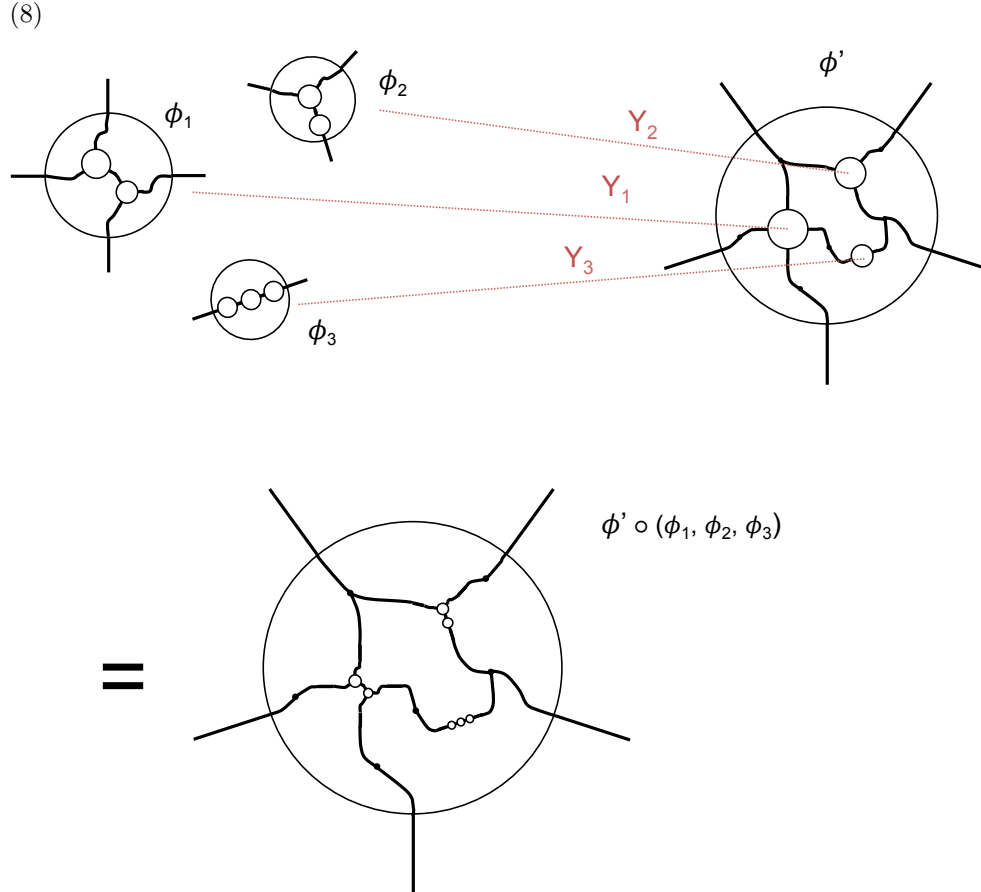
(5)

FIGURE 6. Here is a picture of $\phi\colon (X_1, X_2, X_3) \to Y$. The codomain of ϕ is the outer star $Y = \{a, b, c, d, e\}$ and the $n = 3$ domain objects are the inner stars $X_1 = \{r, s, t\}, X_2 = \{u, v\}$, and $X_3 = \{w, x, y, z\}$. The morphism ϕ has cables $C = \{1, 2, 3, 4, 5, 6\}$, and our picture shows, for example, that $f_1(s) = f_3(w) = g(a) = 1$, i.e. that wires s, w, and a are soldered onto cable 1.

made. This is captured mathematically by the pushout in (7) and is drawn below.

(8)

The identity and associativity law hold by the usual properties of pushouts.

The idea of this paper is to show several applications of the singly-typed wiring diagrams operad as well as a variant, the typed wiring diagrams operad, discussed in Section 4. First we must explain the notion of algebras on an operad.

2.2. **Operad functors and algebras on an operad.** An algebra on an operad is a special case of a functor between operads.

Definition 2.2.1. Let $\mathcal{O}$ and $\mathcal{O}'$ be operads. An *operad functor (*or simply *functor) F from $\mathcal{O}$ to $\mathcal{O}'$*, denoted $F\colon \mathcal{O} \to \mathcal{O}'$, consists of some constituents (A. map on objects, B. map on morphisms) conforming to some laws (1. identity law, 2. composition law). The required constituents for F are as follows:

   A. There is a function $\mathrm{Ob}(F)\colon \mathrm{Ob}(\mathcal{O}) \to \mathrm{Ob}(\mathcal{O}')$.
   B. For each object $y \in \mathrm{Ob}(\mathcal{O})$, finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, and n-indexed set of objects $x\colon n \to \mathrm{Ob}(\mathcal{O})$, there is a function
$$F_n\colon \mathcal{O}_n(x;y) \to \mathcal{O}'_n(Fx;Fy).$$

As in B. above, we often denote $\mathrm{Ob}(F)$, and also each F_n, simply by F. The laws that govern these constituents are as follows:

   1. For each object $x \in \mathrm{Ob}(\mathcal{O})$, the equation $F(\mathrm{id}_x) = \mathrm{id}_{Fx}$ holds.
   2. Let $s\colon m \to n$ be a morphism in **Fin**. Let $z \in \mathrm{Ob}(\mathcal{O})$ be an object, let $y\colon n \to \mathrm{Ob}(\mathcal{O})$ be an n-indexed set of objects, and let $x\colon m \to \mathrm{Ob}(\mathcal{O})$ be an m-indexed set of objects. Then, with notation as in Definition 2.1.2, the following diagram of sets commutes:

$$(9) \qquad \begin{array}{ccc} \mathcal{O}_n(y;z) \times \prod_{i\in n} \mathcal{O}_{m_i}(x_i;y(i)) & \xrightarrow{\ F\ } & \mathcal{O}'_n(Fy;Fz) \times \prod_{i\in n} \mathcal{O}'_{m_i}(Fx_i;Fy(i)) \\ {\scriptstyle \circ}\downarrow & & \downarrow{\scriptstyle \circ} \\ \mathcal{O}_m(x;z) & \xrightarrow[\ \ \ \ \ F\ \ \ \ \ ]{} & \mathcal{O}'_m(Fx;Fz) \end{array}$$

We denote the category of operads and operad functors by **Oprd**.

Warning 2.2.2. What we call operad functors are usually called *operad morphisms* or *morphisms of operads* (see [Lei, Chapter 2]). We found that the notion of morphisms *in an operad* and that of morphisms *of operads* were too easily confused using this terminology. The confusion seems analogous to what would arise if we dismissed the term "functor" altogether from category theory, relying on the term "category morphism" or "morphism of categories" to take over and fulfill its role.

Example 2.2.3. Suppose that **S** and **T** are categories with all finite products. By Example 2.1.5 we have operads **Ss** and **Ts**. If $f\colon \mathbf{S} \to \mathbf{T}$ is a functor that preserves finite products then there is an induced operad functor $F\colon \mathbf{Ss} \to \mathbf{Ts}$. Indeed, on objects we can put $F = f$ since $\mathrm{Ob}(\mathbf{Ss}) = \mathrm{Ob}(\mathbf{S})$ and $\mathrm{Ob}(\mathbf{Ts}) = \mathrm{Ob}(\mathbf{T})$. Every morphism $\phi\colon (x_1,\ldots,x_n;y)$ in **S** is given by a morphism $\phi\colon x_1 \times \cdots \times x_n \to y$ in **S**, so we put

$$F(\phi)\colon F(x_1) \times \cdots \times F(x_n) \cong F(x_1 \times \cdots \times x_n) \xrightarrow{\ f(\phi)\ } F(y).$$

Lemma 2.2.4. *Let $\mathcal{O}$ be an operad, and let $\mathcal{E}$ be the commutative operad from Example 2.1.6. Then there is a unique operad functor $\mathcal{O} \to \mathcal{E}$. In other words, $\mathcal{E}$ is a terminal object in* **Oprd**.

Proof. The operad $\mathcal{E}$ has one object and one morphism in each Hom-set, so the proof is straightforward. □

Definition 2.2.5. Let $\mathcal{O}$ be an operad. An *algebra on $\mathcal{O}$* is an operad functor $R\colon \mathcal{O} \to \mathbf{Sets}$.

Given two algebras $R, S\colon \mathcal{O} \to \mathbf{Sets}$, an *algebra morphism p from R to S*, denoted $p\colon R \to S$, consists of one constituent (A. components) and one law (1. naturality) as follows.

A. For each object $x \in \mathrm{Ob}(\mathcal{O})$ there is a function $p_x\colon R(x) \to S(x)$.

The law that governs the components is as follows:

1. Suppose given an object $y \in \mathrm{Ob}(\mathcal{O})$, a finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, an n-indexed set of objects $x\colon n \to \mathrm{Ob}(\mathcal{O})$, and a morphism $\phi\colon x \to y$. Then the following diagram of sets (the so-called *naturality square*) commutes:

$$
\begin{array}{ccc}
\prod_{i \in n} R(x_i) & \xrightarrow{\;\prod_{i \in n} p_{x_i}\;} & \prod_{i \in n} S(x_i) \\
\downarrow{\scriptstyle R(\phi)} & & \downarrow{\scriptstyle S(\phi)} \\
R(y) & \xrightarrow[\;p_y\;]{} & S(y).
\end{array}
$$

The algebras on $\mathcal{O}$ and their morphisms form a category, which we denote by $\mathcal{O}$–$\mathbf{Alg}$.

Remark 2.2.6. Definition 2.2.5 defines algebra morphisms, but it could easily be modified to define operad natural transformations between arbitrary operad functors $F, G\colon \mathcal{O} \to \mathcal{O}'$. Operad natural transformations $\alpha\colon F \to G$ are defined in the usual way: a component α_x for each object $x \in \mathrm{Ob}(\mathcal{O})$ and naturality squares for each morphism in $\mathcal{O}$. We will not explicitly give the definition in this paper, but it straightforwardly generalizes Definition 2.2.5, in which $\mathcal{O}' = \mathbf{Sets}$.

Example 2.2.7. Let $\mathcal{E}$ be the commutative operad of Example 2.1.6. An $\mathcal{E}$-algebra $S\colon \mathcal{E} \to \mathbf{Sets}$ consists of a set $M \in \mathrm{Ob}(\mathbf{Set})$, and for each natural number $n \in \mathbb{N}$ a morphism $\mu_n\colon M^n \to M$. It is not hard to see that, together the morphism $\mu_2\colon M \times M \to M$ and the element $\mu_0\colon \{*\} \to M$ give M the structure of a commutative monoid. Indeed, the associativity and unit axioms are encoded in the axioms for operads and their morphisms. The commutativity of multiplication arises by applying the commutative diagram (9) to the symmetric group action as discussed in Remark 2.1.3.

Lemma 2.2.8. *Let $\mathcal{O}$ be an operad. The category $\mathcal{O}$–$\mathbf{Alg}$ has all limits and colimits. In particular the initial object sends each $x \in \mathrm{Ob}(\mathcal{O})$ to the empty set, and a terminal object (denoted $\{*\}^{\mathcal{O}}\colon \mathcal{O} \to \mathbf{Sets}$) sends each $x \in \mathrm{Ob}(\mathcal{O})$ to a singleton set.*

Proof. Limits and colimits of $\mathcal{O}$-algebras are formed object-wise on $\mathcal{O}$.

$\square$

The following lemma allows us to transfer algebras on one operad to algebras on another.

Lemma 2.2.9. *Suppose that $F\colon \mathcal{O} \to \mathcal{O}'$ is an operad functor. It induces a functor $F^*\colon \mathcal{O}'$–$\mathbf{Alg} \to \mathcal{O}$–$\mathbf{Alg}$, sending an $\mathcal{O}'$-algebra $i\colon \mathcal{O}' \to \mathbf{Sets}$ to its composition with F. This functor has a left adjoint,*

$$
F_!\colon \mathcal{O}\text{–}\mathbf{Alg} \rightleftarrows \mathcal{O}'\text{–}\mathbf{Alg} \colon\! F^*
$$

Sketch of proof. The functor $F^*\colon \mathcal{O}'$–$\mathbf{Alg} \to \mathcal{O}$–$\mathbf{Alg}$ is clear. The existence of a left adjoint (also known as a left Kan extension) is well-known, but hard to find in the literature. (See [Lur] for a ∞-categorical version.) The construction of the operadic

left Kan extension $F_!$ proceeds analogously to the construction of left Kan extensions in basic category theory. Because we will not use the left adjoint in the current paper, we leave its existence unproven.

$\square$

In this paper we are most concerned with wiring diagrams, and Example 2.2.10 discusses the algebra of relations on the singly-typed wiring diagrams operad $\mathcal{S}$.

Example 2.2.10 (**Main Algebra Example**). Let $\mathcal{S}$ be the singly-typed wiring diagram operad discussed in Example 2.1.7, and fix a set $A \in \mathrm{Ob}(\mathbf{Set})$. In this example we associate to A an $\mathcal{S}$-algebra $\mathcal{R}\mathrm{el}_A$. We call it *the relational $\mathcal{S}$-algebra of type A*.

To specify this $\mathcal{S}$-algebra, i.e. this operad functor $\mathcal{R}\mathrm{el}_A \colon \mathcal{S} \to \mathbf{Sets}$, we need to specify it on objects and on morphisms. Let $X \in \mathrm{Ob}(\mathcal{S}) = \mathrm{Ob}(\mathbf{Fin})$ be a singly-typed star; we assign

(10) $$\mathcal{R}\mathrm{el}_A(X) := \{R \subseteq A^X\} = \mathbb{P}(A^X).$$

That is, $\mathcal{R}\mathrm{el}_A(X)$ is the set of $|X|$-ary relations on A. For any set A, if $X = \varnothing$ then $\mathcal{R}\mathrm{el}_A(X) = \mathbb{P}(\{*\})$ has two elements and can be considered to be boolean valued.

Now suppose that $\phi \colon X_1, \ldots, X_n \to Y$ is a singly-typed wiring diagram given by the cospan to the left below

Note that it induces the diagram to the right above. We need to specify a function

$$\mathcal{R}\mathrm{el}_A(\phi) \colon \mathcal{R}\mathrm{el}_A(X_1) \times \cdots \times \mathcal{R}\mathrm{el}_A(X_n) \to \mathcal{R}\mathrm{el}_A(Y).$$

So suppose given relations $R_1 \subseteq A^{X_1}, \ldots, R_n \subseteq A^{X_n}$. Then $\mathcal{R}\mathrm{el}_A(\phi)(R_1, \ldots, R_n)$ is the relation $S \subseteq A^Y$ defined by first forming the pullback Q in the diagram below,
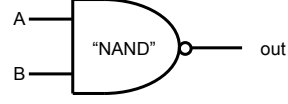
(11)

and then taking as S the image of Q under the induced function $Q \to A^Y$. We refer to $\mathcal{R}\mathrm{el}_A(\phi)(R_1, \ldots, R_n)$ as the *ϕ-conjunction* of relations $R_1, \ldots, R_n$.

Because epi-mono factorizations in **Set** are unique up to isomorphism, it is easy to show that, as specified, $\mathcal{R}\mathrm{el}_A$ respects the composition formula (7) in $\mathcal{S}$.

Example 2.2.11. In the context of digital circuits, it is well known that one can begin with NAND gates and put them together to build NOT gates, AND gates, OR gates, etc. A NAND gate is an element of $\mathcal{R}\mathrm{el}_{\{True, False\}}(\{A,B,out\})$, namely the relation

represented by the table

NAND		
A	**B**	**out**
True	True	False
True	False	True
False	True	True
False	False	True

The wiring diagram ϕ given by the cospan $\{A,B,out\} \xrightarrow{f} \{in,out\} \xleftarrow{g} \{in,out\}$, where $f(A) = f(B) = in$, is depicted by the form of the diagram below. (The interior is "grayed out" for aesthetic reasons, to invoke the idea of viewing NOT as a black box.)
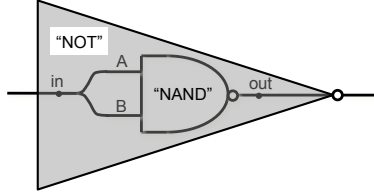

Note however that this diagram depicts more than a morphism $\phi \colon \{A,B,out\} \to \{in,out\}$ in $\mathcal{S}$, but also its application to NAND. That is, it is a depiction of

$$\text{NOT} := \mathcal{R}\mathrm{el}_{\{True,False\}}(\phi)(\text{NAND}),$$

as an element of $\mathcal{R}\mathrm{el}_{\{True,False\}}(\{in,out\})$. One can form relations AND, OR, etc., similarly, and then move on to so-called half adders, full adders, n-bit adders, etc.

At each point the operad allows us to hide the inner implementation details if we choose, or use the composition formula to show the whole picture. We can also regroup different parts of the circuit, and the associative law of operads assures us that this will not change the result.

Example 2.2.12. In this example we return to the three queries from the introduction; see (2). The wiring diagrams for these all involve a three-wire star $\{X,Y,Z\}$, and the first diagram also involves a one-wire star $\{Z\}$.
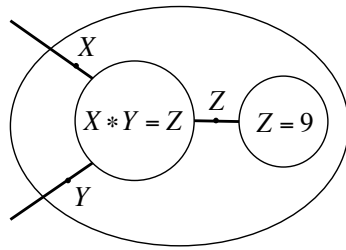
"all pairs of integers whose product is 9"
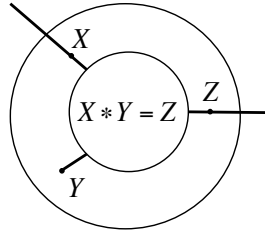
"all pairs of integers in which one is divisible by the other."
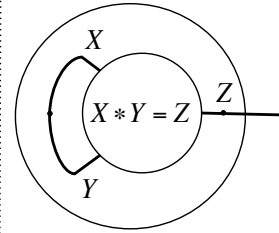
"all perfect squares"

$$\{X,Y\}$$
$$\downarrow g_1$$
$$\{X,Y,Z\} \amalg \{Z\} \xrightarrow{f_1} \{X,Y,Z\}$$

$$\{X,Z\}$$
$$\downarrow g_2$$
$$\{X,Y,Z\} \xrightarrow{f_2} \{X,Y,Z\}$$

$$\{Z\}$$
$$\downarrow g_3$$
$$\{X,Y,Z\} \xrightarrow{f_3} \{XY,Z\}$$

These wiring diagrams are actually morphisms ϕ_1, ϕ_2, ϕ_3 represented by the three cospans above. [4]

For example, we consider $\phi_1 \colon (B_1, B_2) \to C$, where $B_1 = \{X, Y, Z\}$, $B_2 = \{Z\}$, and $C = \{X, Y\}$. The relations $R_1 \in \mathcal{R}el_{\mathbb{N}}(B_1)$ and $R_2 \in \mathcal{R}el_{\mathbb{N}}(B_2)$ are defined by

$$R_1 = \{(X, Y, Z) \in \mathbb{N}^3 \mid X * Y = Z\} \qquad \text{and} \qquad R_2 = \{Z \in \mathbb{N} \mid Z = 9\}.$$

We apply Diagram (11) to our case

$$
\begin{array}{ccc}
 & & \mathbb{N}^{\{X,Y\}} \\
 & & \uparrow \\
\mathbb{N}^{\{X,Y,Z\}} \times \mathbb{N}^{\{Z\}} & \longleftarrow & \mathbb{N}^{\{X,Y,Z\}} \\
\uparrow & & \uparrow \\
R_1 \times R_2 & \longleftarrow & P
\end{array}
$$

where $P = \{(X, Y, Z) \in \mathbb{N}^3 \mid X * Y = Z \ \text{and} \ Z = 9\}$. The ϕ_1-conjunction of R_1 and R_2 is the image of the function $P \to \mathbb{N}^{\{X,Y\}}$, namely

$$\mathcal{R}el_{\mathbb{N}}(\phi)(R_1, R_2) = \{(X, Y) \in \mathbb{N}^2 \mid X * Y = 9\},$$

as desired.

All of these (ϕ_1, ϕ_2, and ϕ_3) can be thought of as conjunctive queries in which the domain is $\mathbb{N}$; we will discuss this further in Section 4.2.

Remark 2.2.13. The $\mathcal{S}$-algebra $\mathcal{R}el_A$ has somewhat counter-intuitive behavior when it comes to inputting empty relations into a wiring diagram. Given a morphism $\phi \colon X_1, \ldots, X_n \to Y$ in $\mathcal{S}$, we can apply the function $\mathcal{R}el_A(\phi)$ to any n-tuple

$$((R_1 \subseteq A^{X_1}), \ldots, (R_n \subseteq A^{X_n})) \in \mathcal{R}el_A(X_1) \times \cdots \times \mathcal{R}el_A(X_n)$$

of relations and return a relation $(S \subseteq A^Y) \in \mathcal{R}el_A(Y)$. The strangeness is that if any of the R_i are empty, then the ϕ-conjunction will also be empty, even if there is no connection between any wire of X_i and a wire of Y.

There are other algebras, related to $\mathcal{R}el$, having different behavior with respect to empty relations. The most intuitive or useful one may be something other than $\mathcal{R}el$, but we decided for clarity to focus on $\mathcal{R}el$ because its construction (see Example 2.2.10) is as straightforward as possible.

Application 2.2.14. The relational $\mathcal{S}$-algebra might be useful in machine learning applications. A goal there could be phrased as follows. Suppose that $R \subseteq A^Y$ is a known $|Y|$-ary relation that we are trying to investigate. An *explanation* of the phenomenon R could be cast as a morphism $\phi \colon X_1, \cdots, X_k \to Y$ and relations $P_1 \subseteq A^{X_1}$ through $P_k \subseteq A^{X_k}$ such that $\mathcal{S}(\phi)(P_1, \ldots, P_k) = R$. We can search through the space of ways to "wire up" known relations to produce the desired relation.

A similar idea should be prevalent as a basic constituent of any design process. The object to be designed is understood as a relationship between its various aspects, and we want to produce it using existing objects, each of which is also a relationship between its various aspects (see [HK]). In other words, the outer relation must be manifest as a wiring diagram of known relations.

---

[4]In ϕ_3 the set C is supposed to have two elements (i.e. XY is the name of one element), and $f_3(X) = f_3(Y) = XY$.

Application 2.2.15. The singly-typed wiring diagram operad $\mathcal{S}$ and its algebras can easily work with time-dependent phenomena. For example, suppose we take $A = C^{\infty}(\mathbb{R})$, the set of smooth functions from $\mathbb{R} \to \mathbb{R}$; we speak as though the independent variable represents time. Relations on A can model such phenomena as

- time delay,
- time-wise addition, subtraction, trigonometric functions, etc., and
- differential equations and differential relations.

For example we can represent a 4-second time delay using a relation $\mathrm{Delay}_4 \subseteq A^2$ where

$$\mathrm{Delay}_4 = \{(f, g) \in A \times A \mid \forall t \in \mathbb{R}, \quad f(t) = g(t + 4)\}.$$

Time-wise addition can be given by the relation $\{(f, g, h) \in C^{\infty}(\mathbb{R}) \mid \forall t \in \mathbb{R}, f(t) + g(t) = h(t)\}$. Similarly, differential equations or differential relations such as $\{(f, g) \in C^{\infty}(\mathbb{R}) \mid \forall t \in \mathbb{R}, (f'(t), g'(t)) \neq (0, 0)\}$.

These are just standard relations. The idea here is that wiring diagrams are a nice formalism for putting them together. To consider functions of more than one variable, one would simply transfer the above ideas to the typed WD operad discussed in Section 4.

3. BRIEF ANALYSIS OF RELATIONAL $\mathcal{S}$-ALGEBRAS

To study the relational $\mathcal{S}$-algebra $\mathcal{R}\mathrm{el}_A$, it would be interesting to find algebraic invariants, i.e. algebra morphisms $\mathcal{R}\mathrm{el}_A \to X$ for some $\mathcal{S}$-algebras X. In other words, suppose we could associate say a complexity level, e.g. a number, to each relation in a database in such a way that the complexity levels add when relations are wired together. This would provide a useful invariant by which we would know, say, the run-time of a query before running it. Unfortunately, it seems that there are very few interesting $\mathcal{S}$-algebra morphisms *out of* $\mathcal{R}$el. Example 3.1.1 and Proposition 3.1.2 explain that there are no interesting additive invariants ("additive" in the sense of the operation of commutative monoids) of $\mathcal{R}$el.

Example 3.1.1. By Example 2.2.7, there is an operad $\mathcal{E}$ such that the category of commutative monoids is equivalent to the category of $\mathcal{E}$-algebras. By Lemmas 2.2.9 and 2.2.4, we can import any $\mathcal{E}$-algebra to an $\mathcal{S}$-algebra by pulling back along the unique functor $t \colon \mathcal{S} \to \mathcal{E}$. However, the resulting $\mathcal{S}$-algebra will not sensitive to the precise wiring pattern but only to the number n of inner stars in a wiring diagram $\phi \colon x_1, \ldots, x_n \to y$. Regardless, if **CMon** is the category of commutative monoids, we have a functor

(12) $$t^* \colon \mathbf{CMon} \to \mathcal{S}\text{–}\mathbf{Alg}.$$

What would it mean to give an algebra morphism $\mathcal{R}\mathrm{el}_A \to t^*(\mathrm{N})$, say if N = $(\mathbb{N}, 0, +)$? In terms of databases (see Example 2.2.12 and Section 4.2) it would mean the following. We would have to assign to every relation a natural number in such a way that when a relation Y is written as a conjunctive query on relations $X_1, \ldots, X_n$, then the number assigned to Y equals the sum of the numbers assigned to the X_i. One way of doing this would be to assign 0 to every relation, but this is uninteresting. Proposition 3.1.2 shows that there generally are no other algebra homomorphisms $\mathcal{R}\mathrm{el}_A \to \mathrm{N}$.

Proposition 3.1.2. *Let A be a set with cardinality $|A| \geqslant 2$, and $\mathcal{R}\mathrm{el}_A$ the A-typed relational $\mathcal{S}$-algebra. Let $(N, 0, +)$ denote any commutative monoid and let $\mathcal{N} = t^*(\mathrm{N}, 0, +)$ denote the $\mathcal{S}$-algebra that assigns the set N to every singly-typed star $X \in$*

$\mathrm{Ob}(\mathcal{S})$ *and summation to morphisms* ϕ *in* $\mathcal{S}$*, as above. The only algebra morphism* $p\colon \mathcal{R}\mathrm{el}_A \to \mathcal{N}$ *is trivial.*

Proof. We will show that p sends every relation to $0 \in N$ in four paragraphs corresponding to the following four steps. First we will show this for any complete relation, then for any non-empty relation, then for the empty 1-ary relation, and finally for the empty relation of any arity.

Let $Y \in \mathrm{Ob}(\mathcal{S})$ be any finite set and let $K_Y \in \mathcal{R}\mathrm{el}_A(Y)$ denote the complete relation $(A^Y \subseteq A^Y)$. Consider the morphism $\phi \in \mathrm{Hom}_{\mathcal{S}}(;Y)$ given by the cospan $\varnothing \to Y \xleftarrow{\mathrm{id}_Y} Y$. Then the image under $\mathcal{R}\mathrm{el}_A(\phi)\colon \{*\} \to \mathcal{R}\mathrm{el}_A(Y)$ of the unique element in the domain is K_Y. On the other hand, the image under $\mathcal{N}(\phi)\colon \{*\} \to \mathcal{N}(Y) = N$ of the unique element in the domain is the empty sum $0 \in N$. By naturality, any algebra morphism $p\colon \mathcal{R}\mathrm{el}_A \to \mathcal{N}$ must assign $p(Y)(K_Y) = 0$.

Now let $X_1, Y \in \mathrm{Ob}(\mathcal{S})$ be any finite sets, and let $(R \subseteq A^{X_1})$ be any element of $\mathcal{R}\mathrm{el}_A(X_1)$ such that $R \neq \varnothing$. Consider the morphism $\psi \in \mathrm{Hom}_{\mathcal{S}}(X_1; Y)$ given by the cospan $X_1 \to X_1 \amalg Y \leftarrow Y$. The image under $\mathcal{R}\mathrm{el}_A(\psi)\colon \mathcal{R}\mathrm{el}_A(X_1) \to \mathcal{R}\mathrm{el}_A(Y)$ of R is again K_Y, whereas the function $\mathcal{N}(\psi)\colon N \to N$ is the identity. By naturality, $p(X_1)(R) = 0$.

Now let $X_1 = X_2 = Y = \{*\}$, and let $a_1 \neq a_2 \in A$ be two distinct elements. Let $R_1 = \{a_1\} \subseteq A^{X_1}$ and let $R_2 = \{a_2\} \subseteq A^{X_2}$. Consider the morphism $\upsilon\colon (X_1, X_2) \to Y$ given by the cospan $\{*\} \amalg \{*\} \to \{*\} \leftarrow \{*\}$. Since R_1 and R_2 are nonempty, but $\mathcal{R}\mathrm{el}(\upsilon)(R_1, R_2) = \varnothing$, we have

$$p(Y)(\varnothing) = p(X_1)(R_1) + p(X_2)(R_2) = 0 + 0 = 0.$$

Finally, let X_1 be any finite set, $Y = \{*\}$. Let $\psi\colon X_1 \to Y$ be given by the cospan $X_1 \to X_1 \amalg Y \leftarrow Y$. If $\varnothing \subseteq A^{X_1}$ is the empty relation then its image under $\mathcal{R}\mathrm{el}_A(\psi)$ is $\varnothing \in \mathcal{R}\mathrm{el}_A(Y)$. As above we have $p(X_1)(\varnothing) = 0$. $\qquad\square$

Proposition 3.1.2 shows that commutative monoids, considered as as $\mathcal{S}$-algebras, are not useful invariants of the $\mathcal{S}$-algebra $\mathcal{R}\mathrm{el}_A$. In fact, non-trivial, non-injective algebra morphisms out of $\mathcal{R}\mathrm{el}_A$ seem to be rare, and this provides a nice challenge to interested readers—see Conjecture 3.1.6. Example 3.1.3 provides another attempt, and Proposition 3.1.4 shows that it fails to be any more informative. However, Example 3.1.3 is interesting in its own right.

Example 3.1.3. In this example, we discuss another $\mathcal{S}$-algebra in which local equivalence relations are wired together to form a global one. Given a finite set $X \in \mathrm{Ob}(\mathcal{S})$, let $Eq(X)$ denote the set of equivalence relations on X. We can think of an equivalence relation $\sim$ on X in terms of the surjection $X \to X/\sim$, and thus depict each equivalence relation on X by connecting the appropriate wires internally. Here is a picture of an equivalence relation $(g \sim h \sim a, b \sim c, d \sim e)$ on $X = \{a, b, c, d, e, f, g, h\}$:
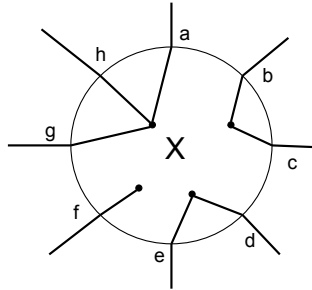
Suppose $\phi\colon (X_1, \ldots, X_n) \to Y$ is a morphism in $\mathcal{S}$ given by the cospan $X_1 \amalg \cdots \amalg$ $X_n \xrightarrow{f} C \xleftarrow{g} Y$. It is easy to see visually how $Eq(\phi)\colon Eq(X_1) \times \cdots \times Eq(X_n) \to Eq(Y)$ should act: if two wires are connected through the wiring diagram, then we consider them to be connected globally.
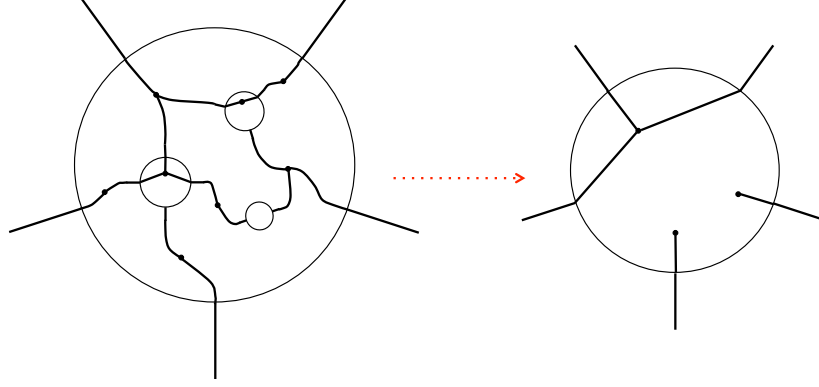

Of course, this can all be done formally. Briefly, given a cospan $\coprod_i X_i \xrightarrow{f} C \xleftarrow{g} Y$ and surjections $X_i \twoheadrightarrow Q_i$ for each $i \in \underline{n}$, we form the surjection $\coprod_i X_i \twoheadrightarrow \coprod_i Q_i$, push it out along f to give some new set P, factor the composite $Y \twoheadrightarrow C \hookrightarrow P$ as a surjection followed by an injection, and take the surjective part $Y \twoheadrightarrow C$ as our induced equivalence relation on Y.

Proposition 3.1.4. *Let $A \in \mathrm{Ob}(\mathbf{Set})$ be a set with cardinality $|A| \geqslant 2$. Let Eq and $\mathcal{R}\mathrm{el}_A$ be the $\mathcal{S}$-algebras discussed above. The only algebra morphism $p\colon \mathcal{R}\mathrm{el}_A \to Eq$ is trivial.*

Proof. The trivial equivalence relation $\Delta_X\colon X \to X \times X$ on a set X is the one corresponding to the surjection $X \xrightarrow{\mathrm{id}_X} X$. For any finite set $X \in \mathrm{Ob}(\mathcal{S})$, consider the morphism $\phi \in \mathrm{Hom}_{\mathcal{S}}(; X)$ given by the cospan $\varnothing \to X \xleftarrow{\mathrm{id}_X} X$. It induces an element $\mathcal{R}\mathrm{el}_A(\phi)(*) \in \mathcal{R}\mathrm{el}_A(X)$ and an element $Eq(\phi)(*) \in Eq(X)$, namely the complete relation K_X and the trivial equivalence relation Δ_X, respectively. It follows by naturality that under any $\mathcal{S}$-algebra morphism $p\colon \mathcal{R}\mathrm{el}_A \to Eq$, we have $p(K_X) = \Delta_X$.

Let $X = \underline{2}$, and let $R \subseteq A^X$ be any relation; we will show that $p(R) = \Delta_X$. For contradiction, assume that $p(R)$ is the only non-trivial equivalence relation on $\underline{2}$, namely the one given by the surjection $m\colon \underline{2} \to \underline{1}$. Let $\psi\colon (\underline{2}, \underline{2}) \to \underline{2}$ be the cospan $\{1_a, 2_a\} \amalg \{1_b, 2_b\} \to \{1_a, 1_b, 2_{ab}\} \leftarrow \{1_a, 1_b\}$ be the wiring diagram as drawn to the left below:
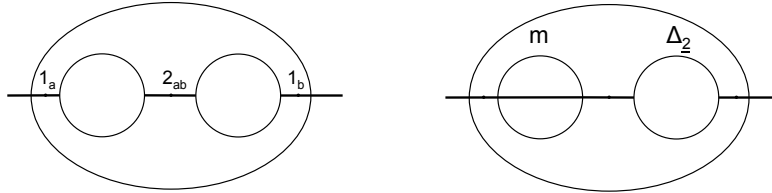
There are two cases. If $R \neq \varnothing$ then $R = \mathcal{R}\mathrm{el}_A(\psi)(R, K_{\underline{2}})$ and (as seen in the right-hand diagram above) we have

$$m = p(R) = p(\mathcal{R}\mathrm{el}_A(\psi)(R, K_{\underline{2}})) = Eq(\psi)(p(R), p(K_{\underline{2}})) = Eq(\psi)(m, \Delta_{\underline{2}}) = \Delta_{\underline{2}},$$

a contradiction. If $R = \varnothing$ we argue as in the proof of Proposition 3.1.2: let $a_1 \neq a_2 \in A$, let $R_1 = \{(a_1, a_1)\}$ and $R_2 = \{(a_2, a_2)\}$. Although both are nonempty, their ψ-conjunction is empty, so we have that

$$m = p(\varnothing) = p(\mathcal{R}\mathrm{el}_A(\psi)(R_1, R_2)) = Eq(\psi)(p(R_1), p(R_2)) = Eq(\psi)(\Delta_{\underline{2}}, \Delta_{\underline{2}}) = \Delta_{\underline{2}},$$

again a contradiction.

Finally, let $X \in \mathrm{Ob}(\mathcal{S})$ be any finite set, let $R' \subseteq A^X$ be any relation, and let $m' := p(X)(R') \in Eq(X)$ be its image under p; we want to show that $m' \colon X \twoheadrightarrow Q$ corresponds to the trivial equivalence relation Δ_X, i.e. that m' is injective. For contradiction, assume m' is not injective, and let $x_1, x_2 \in X$ be such that $m'(x_1) = m'(x_2)$, and consider the morphism $\phi \colon X \to \underline{2}$ given by the cospan $X \xrightarrow{\mathrm{id}_X} X \xleftarrow{(x_1, x_2)} \underline{2}$. Let $R = \mathcal{R}\mathrm{el}_A(\phi)(R')$ be the induced relation and let $m = Eq(\phi)(m')$ be the induced equivalence relation. Note that $m \colon \underline{2} \to \underline{1}$ is nontrivial, and that we have $p(R) = m$, contradicting the work above.

$\square$

It may be that there are no good invariants one can extract from the relational $\mathcal{S}$-algebra. Indeed, after considering the matter for a while we conjecture this to be the case. We make this precise using the following definition.

Definition 3.1.5. Let $\mathcal{O}$ be an operad and let $R \colon \mathcal{O} \to \mathbf{Sets}$ be an algebra. We say that R is *quotient-free* if, for any other $\mathcal{O}$-algebra P and epimorphism $f \colon R \to P$, either f is an isomorphism or P is isomorphic to the terminal $\mathcal{O}$-algebra $\{*\}^{\mathcal{O}}$.

Conjecture 3.1.6. Let $\mathcal{S}$ be the singly typed wiring diagrams operad, let $A \in \mathrm{Ob}(\mathbf{Set})$ be a set, and let $\mathcal{R}\mathrm{el}_A \in \mathrm{Ob}(\mathcal{S}\text{–}\mathbf{Alg})$ be the relational algebra of type A. We conjecture that $\mathcal{R}\mathrm{el}_A$ is quotient-free.

The analysis above is all in the case $|A| \geqslant 2$, whereas this is not assumed in Conjecture 3.1.6. But one can check easily enough that if $|A| \leqslant 1$ then $\mathcal{R}\mathrm{el}_A$ is quotient-free (although note the subtlety that occurs for the object $\varnothing \in \mathrm{Ob}(\mathcal{S})$, e.g. by (10) we have $|\mathcal{R}\mathrm{el}_\varnothing(\varnothing)| = 2$).

4. The typed wiring diagrams operad

The typed wiring diagrams operad is like the untyped version from Example 2.1.7, except that each wire and each cable carries a set of possible values, and these sets must match for wires or cables that are soldered together. In Section 4.1 we will define $\mathcal{T}$ and compare it to $\mathcal{S}$. In Sections 4.2 and 5 we will discuss some applications of both operads.

4.1. **Definition of typed WD operad.**

Example 4.1.1 (**Main Operad Variant**). In this example we discuss the operad $\mathcal{T}$ of typed wiring diagrams. Each object of $\mathcal{T}$ consists of a finite set X and a function $\tau \colon X \to \mathrm{Ob}(\mathbf{Set})$, i.e.

$$\mathrm{Ob}(\mathcal{T}) = \{(X, \tau) \mid X \in \mathrm{Ob}(\mathbf{Fin}), \ \tau \colon X \to \mathrm{Ob}(\mathbf{Set})\} \cong \mathrm{Ob}(\mathbf{Set}/\mathbf{Fin}).$$

We refer to each object as a *typed star*. If $X = \{x_1, x_2, \ldots, x_n)$, we may denote (X, τ) by

$$\{x_1 \colon \tau(x_1), \ x_2 \colon \tau(x_2), \ldots, x_n \colon \tau(x_n)\}.$$

Each $x \in X$ is called a *wire of X*, the set $\tau(x)$ is called the *value type of x*, and each element $v \in \tau(x)$ is called a *value on x*.

We draw an object such as $\{a \colon A, \ b \colon B, \ c \colon C, \ d \colon D, \ e \colon E\}$ as follows:

(13)
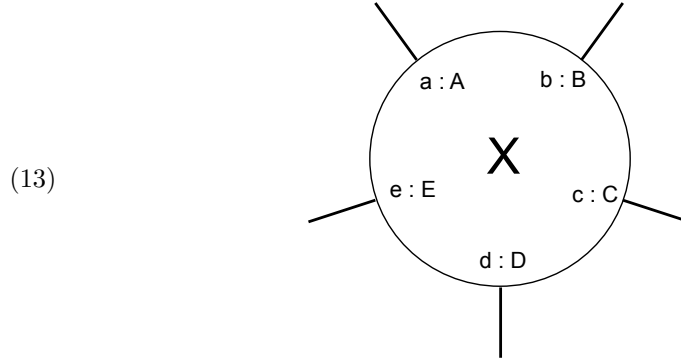
The order of the elements around the circle does not matter: (X, τ) is just a set of five sets.
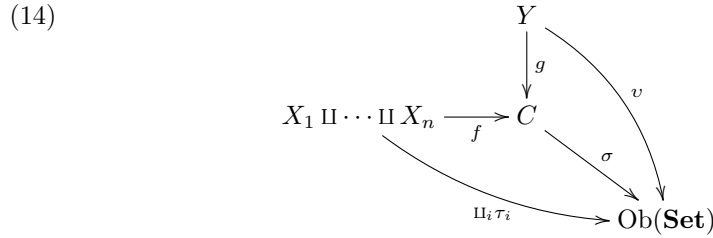
Let $n \in \mathbb{N}$ be a natural number, and let $(Y, \upsilon) \in \mathrm{Ob}(\mathcal{T})$ and $(X_1, \tau_1), \ldots, (X_n, \tau_n) \in \mathrm{Ob}(\mathcal{T})$ be typed stars. A morphism

$$\phi \colon (X_1, \tau_1), \ldots, (X_n, \tau_n) \to (Y, \upsilon)$$

consists of a cospan over $\mathrm{Ob}(\mathbf{Set})$, i.e. a finite set $C \in \mathrm{Ob}(\mathbf{Fin})$, a function $\sigma \colon C \to \mathrm{Ob}(\mathbf{Set})$, and functions f, g as below, making the diagram commute:

(14)

$$
\begin{array}{ccc}
& & Y \\
& & \downarrow{\scriptstyle g} \quad \searrow{\scriptstyle \upsilon} \\
X_1 \amalg \cdots \amalg X_n \xrightarrow{\ f\ } & C & \\
& \searrow{\scriptstyle \sigma} & \\
\xrightarrow{\ \amalg_i \tau_i\ } & & \mathrm{Ob}(\mathbf{Set})
\end{array}
$$

We refer to such a morphism as a *typed wiring diagram*; we refer to the domain objects as the *inner stars*, the codomain object as the *outer star*, and the set C as the *set of cables*. We similarly follow Example 2.1.7 in terms of both notation and depiction, drawing our typed wiring diagrams just like untyped wiring diagrams (5), except perhaps including the typing information (compare (4) and (13)).

Given composable arrows ϕ' and $\phi_1, \ldots, \phi_n$, we can arrange their cospans in the form of a W-shaped diagram and form the pushout as in (7). The pushout dutifully maps to $\mathrm{Ob}(\mathbf{Set})$, so we have the desired composition formula. As in Example 2.1.7, the composition formula is intuitive when drawing pictures; see Diagram (8).

There is a functor $U \colon \mathcal{T} \to \mathcal{S}$ given by forgetting the labelings on each star. Let $A \in \mathrm{Ob}(\mathbf{Set})$ be a set. There is an operad functor $F_A \colon \mathcal{S} \to \mathcal{T}$ given by taking a singly-typed star to the typed-star with every wire having type A. The composite

$$\mathcal{S} \xrightarrow{\ F_A\ } \mathcal{T} \xrightarrow{\ U\ } \mathcal{S}$$

is the identity on $\mathcal{S}$.

Lemma 4.1.2. *For any set $A \in \mathrm{Ob}(\mathbf{Set})$, let $F_A \colon \mathcal{S} \to \mathcal{T}$ be the morphism given above and let $F_A^* \colon \mathcal{T}\text{–}\mathbf{Alg} \to \mathcal{S}\text{–}\mathbf{Alg}$ be the functor from Lemma 2.2.9, and let $\mathcal{R}\mathrm{el}_A \in \mathcal{S}\text{–}\mathbf{Alg}$ be the algebra given in Example 2.2.10. There exists a $\mathcal{T}$-algebra $\mathcal{R}\mathrm{el} \colon \mathcal{T} \to \mathbf{Sets}$ such that for every set A we have an isomorphism of $\mathcal{S}$-algebras*

$$F_A^*(\mathcal{R}\mathrm{el}) \cong \mathcal{R}\mathrm{el}_A.$$

Proof. We simply follow Example 2.2.10 in defining *the relational $\mathcal{T}$-algebra*, $\mathcal{R}\mathrm{el}\colon \mathcal{T} \to$ **Sets**. Given a typed star $\overline{X} = (X, \tau) \in \mathrm{Ob}(\mathcal{T})$, let $\Pi_X(\tau)$ denote the set

$$\Pi_X(\tau) := \prod_{x \in X} \tau(x).$$

We assign $\mathcal{R}\mathrm{el}(\overline{X}) = \{R \subseteq \Pi_X(\tau)\}$, the set of *relations of type $\overline{X}$*.

Suppose given a typed wiring diagram $\phi\colon (X_1, \tau_1), \cdots, (X_n, \tau_n) \to (Y, \upsilon)$ as in (14) and relations $R_i \in \Pi_{X_i}(\tau_i)$ for each $i \in \underline{n}$. Analogous with the singly-typed case (11) we form the pullback diagram

$$
\begin{array}{ccccc}
 & & & & \Pi_Y(\upsilon) \\
 & & & & \uparrow {\scriptstyle \Pi_g} \\
\Pi_{X_1}(\tau_1) \times \cdots \times \Pi_{X_n}(\tau_n) & \xleftarrow{\;\Pi_f\;} & \Pi_C(\sigma) & & S \\
\uparrow & & \uparrow {\scriptstyle \ulcorner} & & \uparrow \\
R_1 \times \cdots \times R_n & \longleftarrow & & Q &
\end{array}
$$

and take as S the image of Q under the induced function $Q \to \Pi_Y(\upsilon)$. We define

$$\mathcal{R}\mathrm{el}(\phi)(R_1, \ldots, R_n) := S$$

and refer to it as the *ϕ-conjunction* of $R_1, \ldots, R_n$. It is easy to see that, under this specification, $\mathcal{R}\mathrm{el}$ respects the composition formula for $\mathcal{T}$ and thus is a $\mathcal{T}$-algebra.

To see that $F_A^*(\mathcal{R}\mathrm{el}) \cong \mathcal{R}\mathrm{el}_A$ is straightforward by construction.

$\square$

4.2. **Databases using the typed wiring diagrams operad.** The so-called *logical structure of a relational database* can be defined as follows. At the highest level, the logical structure consists of a finite set P of predicates, each of which will eventually correspond to (i.e. holds the place for) a table. A *predicate p* consists of a finite set of attributes, each of which is a set, i.e. $p = (n_p, A_p)$, where $n_p \in \mathrm{Ob}(\mathbf{Fin})$ is a finite set and $A_p\colon n_p \to \mathrm{Ob}(\mathbf{Set})$. We sometimes denote p by $(A_p(1), \ldots, A_p(n_p))$. The set n_p corresponds to the set of columns of p's table, and for each column $i \in n_p$, the set $A_p(i)$ is the set of values that can occur in that column. Some relational databases do not allow for different data types. In such cases there is a single universe $U \in \mathrm{Ob}(\mathbf{Set})$ and $A_p(i) = U$ for each $i \in n_p$,

Given such a logical structure for a relational database, an *instance* of it consists of a relation $R_p \subseteq A_p(1) \times \cdots \times A_p(n_p)$ for each predicate $p \in P$. Being in the presence of an instance means that the tables we looked forward to above have been filled in with data of the appropriate types.

A *query* is roughly a question asked of the database. Given a set P of predicates as above, a *conjunctive query* is often written in the following form

SELECT	$x_1, x_2, \ldots, x_k$
FROM	$p_1, p_2, \ldots, p_\ell$
WHERE	$x_{i_1} = x'_{i_1}$ AND $x_{i_2} = x'_{i_2}$ AND $\cdots$ AND $x_{i_m} = x'_{i_m}$

Here each x_i (and x_{i_j} and x'_{i_j}) refers to a specific attribute in a specific predicate $p_1, \ldots, p_\ell$. A conjunctive query can also be thought of as follows. Start with a set of relations and take their Cartesian product (using the FROM-clause), then choose those rows for which the values in specified pairs of columns agree (using the WHERE-clause), and then project down to a subset of the columns (using the SELECT-clause).

Wiring diagrams (i.e. morphisms in $\mathcal{T}$) give us a way to picture conjunctive queries. For example, a predicate $p = (n_p, A_p)$ is precisely the same thing as a typed star, i.e. an object $p \in \mathrm{Ob}(\mathcal{T})$. [5] A morphism $\phi\colon ((n_1, A_1), \ldots, (n_\ell, A_\ell)) \to (k, x)$ consists of a cospan over **Set**, i.e. a diagram of the form

$$
\begin{array}{ccc}
 & k & \\
 & \downarrow g & \\
 & & \searrow^{x} \\
n_1 \amalg \cdots \amalg n_\ell \xrightarrow{f} C & & \\
 & \searrow_{A} & \\
 & & \\
\amalg_i A_i \longrightarrow & \mathrm{Ob}(\mathbf{Set})
\end{array}
$$

Conjunctive queries are a bit more limited than what can be expressed using wiring diagrams. This will be explored a bit in Example 4.2.2.

Example 4.2.1. Here is an example of a conjunctive query from wikipedia, written in the database query language SQL. It uses constants ('male' and 'female'), whereas our above description of conjunctive queries does not, but we can still make good sense of it with wiring diagrams.

SELECT	L.student, L.address
FROM	attends a1, gender g1, attends a2, gender g2, lives L
WHERE	a1.student=g1.student AND a2.student=g2.student
AND	L.student=g1.student AND a1.course=a2.course
AND	g1.gender='male' AND g2.gender='female'

Below in (15) we show this query in the form of a wiring diagram. The five black inner stars correspond to the five tables in the FROM clause; the two red inner stars are the constants; the two wires of the outer star correspond to the two attributes in the SELECT clause; and the precise wiring pattern corresponds to the conditions in the WHERE clause.
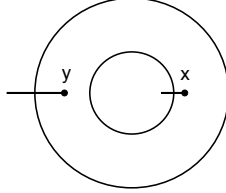
(15)
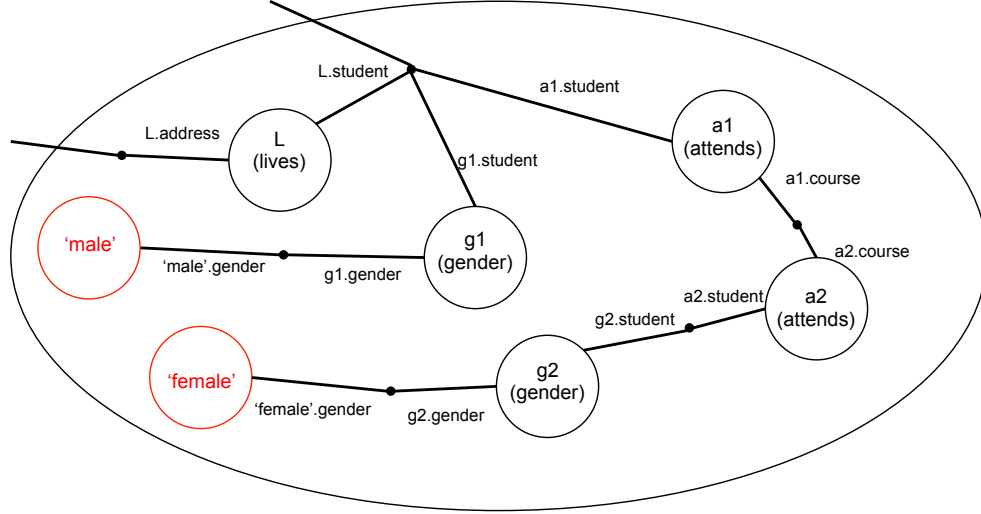
The constants 'male' and 'female' are each simply considered as a 1-ary relation on the set of genders.

Example 4.2.2. Here we show a non-conjunctive query that can be made using a wiring diagram, namely $y.\exists x.R(x)$ for some unary relation R. The wiring diagram ϕ is represented by the cospan is $\{x\} \xrightarrow{f} \{x,y\} \xleftarrow{g} \{y\}$,

and the reason it is considered not to be conjunctive is that y is not in the image of f.

Given a relation $R \in \mathcal{R}\text{el}(\{x\})$, the function $\mathcal{R}\text{el}(\phi)\colon \mathcal{R}\text{el}(\{x\}) \to \mathcal{R}\text{el}(\{y\})$ returns $(\varnothing \subseteq S)$ if $R = \varnothing$ and returns $(S \subseteq S)$ if $R \neq \varnothing$. (See Remark 2.2.13). That is $\mathcal{R}\text{el}(\phi)(R)$ is $y.\exists x.R(x)$.

4.3. **Disjunctive queries.** We can add disjunctive queries to our operadic query language by considering $\mathcal{R}\text{el}$ as an $\mathcal{T}$-algebra object in **JLats**, the operad of join semi-lattices, rather than in **Sets**. See Example 2.2.3. To make this precise, we have the following proposition.

Proposition 4.3.1. *Let $\mathcal{T}$ be the typed wiring diagrams operad, and let $\mathcal{R}\text{el}\colon \mathcal{T} \to$ **Sets** be the relational $\mathcal{T}$-algebra. Let $U\colon$ **JLats** $\to$ **Sets** be the operad functor that sends a join-semilattice to its underlying set. Then there exists a morphism $J\mathcal{R}\text{el}\colon \mathcal{T} \to$ **JLats** such that the following diagram of operads commutes:*

$$\mathcal{T} \xdashrightarrow{\ J\mathcal{R}\text{el}\ } \mathbf{JLats}$$

$$\mathcal{R}\text{el} \searrow \qquad \swarrow U$$

$$\mathbf{Sets}.$$

Proof. Given a typed star $\overline{X} := (X, \tau) \in \text{Ob}(\mathcal{T})$, where $X \in \text{Ob}(\mathbf{Fin})$ and $\tau\colon X \to \text{Ob}(\mathbf{Sets})$, we define the join-semilattice

$$J\mathcal{R}\text{el}(X,\tau) := \{R \subseteq \prod_{x \in X} \tau(x)\}, \qquad \text{with} \qquad \bigvee_{i \in I} R_i := \bigcup_{i \in I} R_i.$$

The underlying set $U \circ J\mathcal{R}\text{el}(X,\tau)$ is $\mathcal{R}\text{el}(X,\tau)$, so on objects, the triangle commutes. We need to say what $J\mathcal{R}\text{el}$ does on morphisms $\phi\colon \overline{X}_1, \ldots, \overline{X}_n \to \overline{Y}$ in $\mathcal{T}$.

Given a cospan $(X,\tau) \xrightarrow{f} (C,\sigma) \xleftarrow{g} (Y,\upsilon)$, as in Diagram (14), we have a span of sets

$$\prod_X \tau \xleftarrow{\Pi_f} \prod_C \sigma \xrightarrow{\Pi_g} \prod_Y \upsilon.$$

The prescription for $\mathcal{R}\text{el}(\phi)$ says that for any relation $R \subseteq \prod_X \tau$, we pull it back to a relation on $\prod_C \sigma$ and then take its image in $\prod_Y \upsilon$. We need only to see that this process preserves the join, i.e. union of subsets. But both pullback and image are left adjoints; since unions are colimits, they are preserved.

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

Using algebra objects in **JLat**, as in Proposition 4.3.1, we can take the union of any two relations if they have the same type (X, τ). Doing so will appropriately commute with ϕ-conjunction.

5. Closed operads for plug-and-play and recursion

5.1. **Plug-and-play.** Hot-plugging, or a variant called *plug-and-play*, is a computer engineering term, which refers to the capacity whereby one or more components of a computer system can be replaced without interruption to the system. We use the term only slightly differently here, in that we are not considering computers per se but any system that can be described solely by a wiring diagram. As such, rebooting is not the issue. Instead the issue is to build systems that have slots in which components can be filled in later.

To make sense of this idea, we will use the notion of closed operads. The term should be understood in relation to Cartesian closed categories (CCCs), like those used as a denotational semantics for computer programs. The similarity between CCCs and closed operads is found in the ability to *curry* a morphism f with multiple input types, i.e. to translate f into an equivalent function that has fewer inputs but whose output is a function on the remaining inputs. The difference between closed operads and CCCs is that closed operads literally have multiple inputs, whereas CCCs in some sense *simulate* having multiple inputs, using Cartesian product. Since **Set** is a Cartesian closed category, the relationship between operads and CCCs is captured using algebras on an operad $\mathcal{O}$, i.e. operad functors $\mathcal{O} \to \mathbf{Sets}$ by which the multiple input types of an $\mathcal{O}$-morphism are translated into the respective Cartesian product.

The following definitions and terminology follows [Man].

Definition 5.1.1. A *closed operad*, denoted $(\mathcal{O}, \Rightarrow, ev)$, consists of three constituents (A. operad, B. internal Hom-objects, C. evaluation map) conforming to one law (1. externalization). The required constituents are as follows:

   A. There is a operad $\mathcal{O}$.
   B. For each object $z \in \mathrm{Ob}(\mathcal{O})$, finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, and n-indexed set of objects $y \colon n \to \mathrm{Ob}(\mathcal{O})$, there is an object $[y \Rightarrow z] \in \mathrm{Ob}(\mathcal{O})$ called *the internal Hom-object associated to y and z*.
   C. For each object $z \in \mathrm{Ob}(\mathcal{O})$, finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, and n-indexed set of objects $y \colon n \to \mathrm{Ob}(\mathcal{O})$, there is an $(n+1)$-ary evaluation morphism
   $$ev_{y;z} \colon ([y \Rightarrow z], y) \longrightarrow z$$

We may denote the object $[y \Rightarrow z]$ by $[(y_1, \ldots, y_n) \Rightarrow z]$, when convenient or by $[y \Rightarrow z]_{\mathcal{O}}$ if necessary. The law that governs these constituents is as follows:

   1. Suppose given an object $z \in \mathrm{Ob}(\mathcal{O})$, finite sets $m, n \in \mathrm{Ob}(\mathbf{Fin})$, and so-indexed sets of objects $x \colon m \to \mathrm{Ob}(\mathcal{O})$ and $y \colon n \to \mathrm{Ob}(\mathcal{O})$. The function
   $$extl_{x,y;z} \colon \mathcal{O}_m(x; [y \Rightarrow z]) \longrightarrow \mathcal{O}_{m+n}(x, y; z),$$
   defined by sending a morphism $\phi \colon x \to [y \Rightarrow z]$ to the composite
   $$x_1, \ldots, x_m, y_1, \ldots, y_n \xrightarrow{\phi, \mathrm{id}_{y_1}, \ldots, \mathrm{id}_{y_n}} [(y_1, \ldots, y_n) \Rightarrow z], y_1, \ldots, y_n \xrightarrow{ev_{y;z}} z,$$
   is bijective.

The isomorphism $extl_{x,y;z}$ defined above is called the *externalization function*; we denote its inverse by $intl_{x,y;z}$ and call it the *internalization function*.

A *functor* between closed operads is just a functor between their underlying operads, and an *algebra* on a closed operad is just an algebra on the underlying operad.

Remark 5.1.2. The categorically-minded reader may notice that Definition 5.1.1 does not explicitly require that the internal Hom-objects $[X \Rightarrow Y]$ be functorial in X or Y. However, this functoriality follows from their tight relation to the actual hom-sets, $\mathcal{O}(X;Y)$, which are functorial in X and Y by definition.

Example 5.1.3. The operad **Sets** of sets is closed. Indeed, given sets $Y_1, \ldots, Y_n, Z$ we can define the internal-Hom object $[(Y_1, \ldots, Y_n) \Rightarrow Z] \in \mathrm{Ob}(\textbf{Sets})$ to be the set $\mathrm{Hom}_{\textbf{Sets}}(Y_1, \ldots, Y_n; Z)$.

Proposition 5.1.4. *The singly-typed wiring-diagram operad $\mathcal{S}$ (respectively, the typed wiring-diagram operad $\mathcal{T}$) is closed.*

Proof. We prove this for $\mathcal{S}$ because it is less cluttered, but the proof for $\mathcal{T}$ is perfectly analogous, adding a map from everything in sight to $\mathrm{Ob}(\textbf{Set})$, as in Example 4.1.1.

Suppose given finite sets $Y_1, \ldots, Y_n, Z \in \mathrm{Ob}(\mathcal{S})$. We define the associated internal Hom-object using the coproduct,

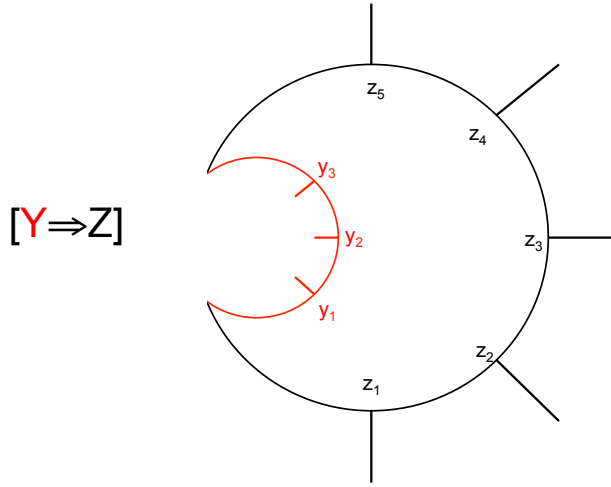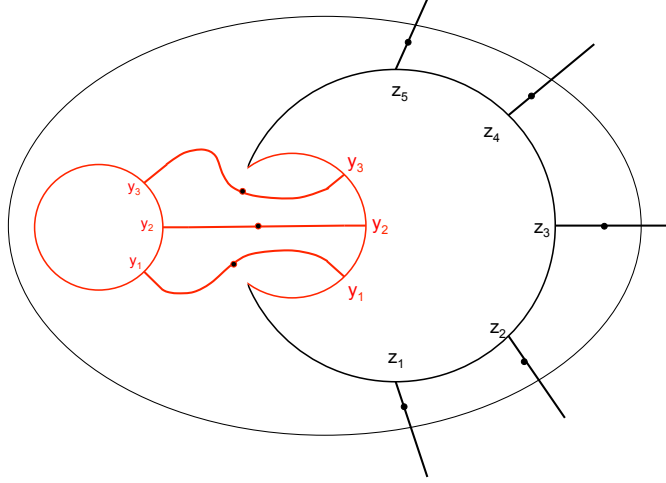$$[(Y_1, \ldots, Y_n) \Rightarrow Z] := Y_1 \amalg \cdots \amalg Y_n \amalg Z.$$

FIGURE 16. A picture of $[Y_1 \Rightarrow Z]$, where $n = 1$, $Y_1 = \{y_1, y_2, y_3\}$, and $Z = \{z_1, z_2, z_3, z_4, z_5\}$. The shape of the picture is meaningless; it is merely a suggestive depiction of the set $[Y_1 \Rightarrow Z] = Y_1 \amalg Z$.

The evaluation morphism $ev_{Y;Z} \in \mathcal{S}_{n+1}([Y \Rightarrow Z], Y; Z)$ is the following cospan

$$\left(\coprod_{i=1}^{n} Y_i \right) \amalg Z \amalg \left(\coprod_{i=1}^{n} Y_i \right) \longrightarrow \left(\coprod_{i=1}^{n} Y_i \right) \amalg Z \longleftarrow Z,$$

where the first function is given by the fold map. Here is a picture of the evaluation map in the above case:

To see that $(\mathcal{S}, \Rightarrow, ev)$ satisfies the externalization condition, we begin with finite sets $X_1, \ldots, X_m, Y_1, \ldots, Y_n, Z \in \mathrm{Ob}(\mathcal{S})$ and a morphism $\phi \colon X \to [Y \Rightarrow Z]$ given by the cospan $X \xrightarrow{f} C \xleftarrow{g} [Y \Rightarrow Z]$. We must show that the externalization function $extl_{X,Y;Z} \colon \mathcal{S}_m(X; [Y \Rightarrow Z]) \to \mathcal{S}_{m+n}(X, Y; Z)$ is a bijection, in accordance with Definition 5.1.1. By definition, the morphism $extl_{X,Y;Z}(\phi) \colon (X, Y) \to Z$ is given by forming the following diagram

$$
\begin{array}{ccc}
 & & Z \\
 & & \downarrow \\
[Y \Rightarrow Z] \amalg Y & \xrightarrow{\ fold_Y\ } & [Y \Rightarrow Z] \\
{\scriptstyle g, \mathrm{id}_Y} \downarrow & & \downarrow \\
X \amalg Y \xrightarrow{\ f, \mathrm{id}_Y\ } C \amalg Y & \xrightarrow{\qquad} & C
\end{array}
$$

The bottom-right object is, according to the composition law for $\mathcal{S}$, a pushout; it is easy to check set-theoretically that this pushout is isomorphic to C, as written. The morphism $extl_{X,Y;Z}(\phi)$ is represented by the cospan $X \amalg Y \to C \leftarrow Z$ obtained by composites. However an easier way to understand what is going on is simply that $extl_{X,Y;Z}$ converts diagrams like the one on the left below to diagrams like the one on the right:

$$
\begin{array}{ccc}
Y \amalg Z & & & & Z \\
{\scriptstyle g_1, g_2} \downarrow & & \xrightarrow{\ extl_{X,Y;Z}\ } & & \downarrow {\scriptstyle g_2} \\
X \xrightarrow{\ f\ } C & & & & X \amalg Y \xrightarrow{\ f, g_1\ } C
\end{array}
$$

It is clear that this correspondence ($extl_{X,Y;Z}$) is a bijection, so $\mathcal{S}$ is indeed closed.

$\square$

Remark 5.1.5. We will show that in $\mathcal{S}$ (with an eye toward its algebra $\mathcal{R}$el), one can speak of a morphism to the internal Hom-object $[(Y_1, \ldots, Y_n) \Rightarrow Z]$ as an *explanation of Z, specifically in terms of $Y_1, \ldots, Y_n$*. If this is to be the case then when $n = 0$ we must have that morphisms to $Z \cong [() \Rightarrow Z]$ are explanations of Z (in no specific terms). This makes sense: a morphism $(X_1, \ldots, X_n) \to Z$ is a wiring diagram, which explains the behavior of the outer relation in terms of, i.e. as a query on, the

inner relations. Similarly, a morphism $X \to [Y \Rightarrow Z]$ is the same as a morphism $(X, Y) \to Z$, so it is an explanation of Z that in particular involves the star Y.

Definition 5.1.6. Suppose that $F \colon \mathcal{O} \to \mathcal{O}'$ is a morphism of closed operads. For any object $Y \in \mathrm{Ob}(\mathcal{O})$, finite set $n \in \mathrm{Ob}(\mathbf{Fin})$, and n-indexed set of objects $X \colon n \to \mathrm{Ob}(\mathcal{O})$, let $ev_{X;Y}$ be the evaluation morphism in $\mathcal{O}$. Applying F we obtain an $\mathcal{O}'$-morphism $F(ev_{X;Y}) \colon (F[X \Rightarrow Y]_{\mathcal{O}}, FX) \to FY$. Define the *closing transformation* of F to be the $\mathcal{O}'$-morphism

$$(17) \qquad F_{X;Y} \colon F[X \Rightarrow Y]_{\mathcal{O}} \longrightarrow [FX \Rightarrow FY]_{\mathcal{O}'},$$

obtained from $F(ev_{X;Y})$ by applying the internalization function.

5.2. **Recursion.** We can think of recursion as what happens when a relation (e.g. a function) is "plugged into itself". The story will go as follows. We use the fact that $\mathcal{S}$ is a closed operad and define a *recursive star* to be an object of the form $[Z \Rightarrow Z] \in \mathrm{Ob}(\mathcal{S})$. A *recursive setup* will be any wiring diagram $\phi \colon (X_1, \ldots, X_n) \to [Z \Rightarrow Z]$ with output $[Z \Rightarrow Z]$, and with the input stars filled in with relations $R_1 \in \mathcal{R}\mathrm{el}(X_1), \ldots, R_n \in \mathcal{R}\mathrm{el}(X_n)$. Finally, a *recursive relation* for this setup will be any (non-empty) fixed point of the resulting function

$$\phi(R_1, \ldots, R_n) \colon \mathcal{R}\mathrm{el}(Z) \to \mathcal{R}\mathrm{el}(Z).$$

The following picture of a recursive setup, for which the factorial function is a fixed point, may be helpful.

(18)
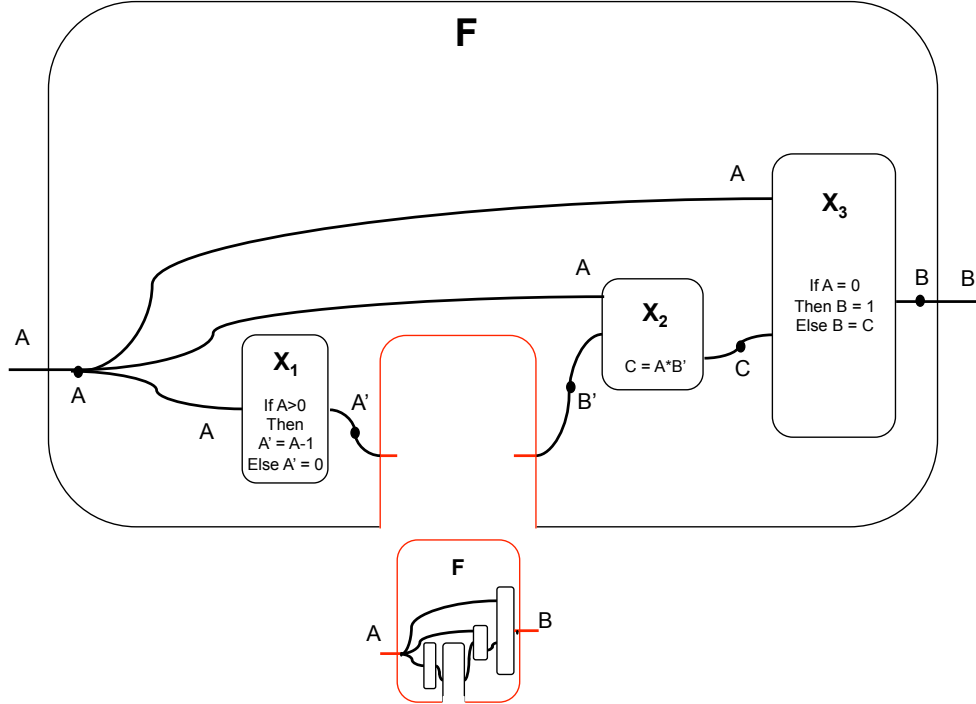
We will now describe this picture in an example.

Example 5.2.1. First let us run through the recursive procedure being depicted in (18). We are describing a relation, which is supposedly the factorial function, in terms of some other relations and itself. The idea is that given a nonzero natural

number A, we subtract 1, take the factorial $(A-1)!$, multiply this number by A, and output the result. If $A = 0$, output 1.

The diagram shown above consists of two parts: a morphism $\phi \colon (X_1, X_2, X_3) \to [F \Rightarrow F]$ in $\mathcal{S}$ and some chosen relations. Here $X_1 := \{A, A'\}, X_2 := \{A, B', C\}, X_3 := \{A, C, B\}$, and $F := \{A, B\}$. The recursive star is $[F \Rightarrow F] \cong \{A, A', B, B'\}$. The morphism ϕ is represented by the cospan

$$\phi \colon \{A, A'\} \amalg \{A, B', C\} \amalg \{A, C, B\} \longrightarrow \{A, A', B, B', C\} \longleftarrow \{A, A', B, B'\}.$$

The chosen relations are written in the boxes. They are:

Decrement: " If $A > 0$ then $A' = A - 1$ else $A' = 0$ " as an element $R_1 \in \mathcal{R}el_{\mathbb{N}}(X_1)$;

Multiplication: " $C = A * B'$ " as an element $R_2 \in \mathcal{R}el_{\mathbb{N}}(X_2)$; and

Conditional: " If $A = 0$ then $B = 1$ else $B = C$ " as an element $R_3 \in \mathcal{R}el_{\mathbb{N}}(X_3)$.

Together these constitute an element (R_1, R_2, R_3) of the left-hand side:

$$\mathcal{R}el_{\mathbb{N}}(X_1) \times \mathcal{R}el_{\mathbb{N}}(X_2) \times \mathcal{R}el_{\mathbb{N}}(X_3) \xrightarrow{\phi} \mathcal{R}el_{\mathbb{N}}([F \Rightarrow F])$$

This completes the recursive setup. From it we can form the ϕ-conjunction,

$$\mathcal{R}el_{\mathbb{N}}(\phi)(R_1, R_2, R_3) \in \mathcal{R}el_{\mathbb{N}}([F \Rightarrow F]),$$

and applying to it the closing transformation (17), we have a function

$$q \colon \mathcal{R}el_{\mathbb{N}}(F) \to \mathcal{R}el_{\mathbb{N}}(F).$$

In terms of the diagram above, q is the function that takes a relation on $\mathbb{N} \times \mathbb{N}$, plugs it into the empty slot, and returns a relation on $\mathbb{N} \times \mathbb{N}$. A recursive relation for this setup is a non-empty fixed point of q. It turns out there is only one: the factorial relation, $B = A!$.

6. Future work

There are three main directions for taking this project further. The first is to investigate the mathematical properties of the wiring diagrams operad $\mathcal{S}$ (or $\mathcal{T}$) and its algebras; this is discussed in Section 6.1. The second is to use the wiring diagrams operad and its algebras to study real-world phenomena; this is discussed in Section 6.2. The third is to change and expand the model, i.e. the operad $\mathcal{S}$ itself; this is discussed in Section 6.3.

6.1. **Investigating mathematical properties of the WD operad and its algebras.** It seems difficult to find any simplifying invariants of relations that are natural with respect to wiring diagrams. This was the content of Conjecture 3.1.6. It would be quite interesting to know definitively whether such an invariant can exist. It would also be good to know whether relational $\mathcal{S}$-algebras have any privileged place in $\mathcal{S}$–**Alg**, e.g. whether the algebras $\mathcal{R}el_A$ can be characterized category-theoretically. Many such mathematical questions remain open about relational $\mathcal{S}$-algebras.

It is also possible that we could greatly decrease the number of morphisms in each hom-set of $\mathcal{S}$ without having much affect on the category of algebras. For example, there is a sub-operad $\mathcal{S}' \subseteq \mathcal{S}$ in which each wiring diagram is required to have exactly one connected component, an upshot of which is that all hom-sets in $\mathcal{S}'$ are finite. We have not yet investigated how the category $\mathcal{S}'$–**Alg** compares to $\mathcal{S}$–**Alg**.

The wiring diagrams operad $\mathcal{S}$ itself has more structure than we have considered here. First of all, the objects of $\mathcal{S}$ are finite sets, but we never discuss how functions between them enter the picture. Second, the morphisms $\phi \colon X \to Y$ in $\mathcal{S}$ are cospans in **Fin** of the form $X \to C \leftarrow Y$, but we never discuss the fact that there is a whole

category of such morphisms (i.e. we can discuss morphisms of cospans). In other words, $\mathcal{S}$ can perhaps be lifted to something like an operad object in **Cat**.

It would also be useful to investigate whether aggregation, such as taking sums or averages in a database, can be understood in terms of the wiring diagrams operad $\mathcal{S}$ or $\mathcal{T}$, perhaps using the fact that they are closed. That is, we have a way to speak about functions from relations to relations, and aggregation may be phrased in this way.

In that vein, it would be valuable to formalize more of relational database theory, including natural keys, normal forms, negation, nulls, queries etc., in the language of operads.

6.2. **Using the WD operad and various algebras to study real-world phenomena.** One of the most important questions, in the author's opinion, is how structure and function are related in the human brain. If the brain is understood as a collection of neurons wired together at synapses, then we are tasked with explaining how complex functionality like walking and calculating can be housed in such a structure. The wiring diagrams operad can serve as a language for discussing the structure of the brain, in the sense that we need not think of neurons as indivisible objects, nor do we need to step from the neuron level directly to the brain level. The brain can be divided into different agencies, such as the visual cortex and the auditory cortex, and each of these can be broken down further. The WD operad allows us to zoom into any given part (either topological region or functional subset) and then connect our findings about it with those about other regions.

Our findings about each region, be it neuron, whole brain, or anything in between, might be phrased in terms of algebras over the operad $\mathcal{S}$. In this paper we used the relational $\mathcal{S}$-algebra, which would be relevant if we think of each portion of the brain as managing a relationship between its inputs and outputs. But in general, choosing an algebra allows us to do two things. First, we get to assign to each agency of the brain a set of possible states, and second, we get to assign a rule by which a state on any given whole emerges from the states on its constituent parts.

Besides using the WD operad to discuss the brain, we may also find it useful in understanding the process of design. It seems that intelligent animals can form a concept of what they want to see emerge in the world, and then work to create it. This requires a constant back-and-forth between what is known and what is intended (see [HK]). We build a whole new object with certain properties by putting together known objects in creative patterns. The wiring diagrams operad or something related could be useful to frame the discussion, because it carefully delineates between the wiring pattern (operad), and how large-scale function emerges from small-scale function via that wiring pattern (algebra).

6.3. **Expanding the model.** The wiring diagrams operad in this paper provides a simple picture for how smaller entities can be put together to form larger ones, but it is certainly not the only operad that would fit such a description. For example, a directed version of the wiring diagrams operad with an algebra of functions has been shown to exist and a paper on the subject [SS] is in preparation. Directed wiring diagrams may be useful in describing processes, or recipes, whereby one process (taking inputs to outputs) is made by an ordered sequence of steps (each of which involving its own inputs and outputs).

We may also want to expand the wiring diagrams operad so that the actual spacing between inner stars, or the length of the wires, becomes part of the equation, so to speak. For example, the classical "little cubes operad" would become a suboperad of

such a *topological wiring diagrams* operad. The length of wires could encode Shannon-style information theory concepts, giving rise to a richer theory.

In a totally different direction, one could study the mechanical properties of topological wiring diagrams: if we know how each inner star deforms under stress, and we know how the outer star is composed of inner stars, then perhaps we can calculate how the outer star will deform under stress. This framework could prove useful to materials scientists.

References

[AHV]   Abiteboul, S.; Hull, R.; Vianu, V. (1995) *Foundations of databases.* Addison Wesley.
[Awo]   S. Awodey. (2010) *Category theory.* Second edition. Oxford Logic Guides, 52. Oxford University Press, Oxford.
[BV]   Boardman, M.; Vogt, R. (1973) "Homotopy invariant algebraic structures on topological spaces." *Lecture notes in mathematics* 347. Springer-Verlag.
[BW]   M. Barr, C. Wells. (1990) *Category theory for computing science.* Prentice Hall International Series in Computer Science. Prentice Hall International, New York.
[Eme]   Emery, R.C. (1985) *Digital circuits: Logic and design.* CRC Press.
[HK]   Hendriks, L.; Kazakci, A. (2010) "A formal account of the dual expansion of concepts and knowledge in C-K theory." In *Proceedings of the 11th international design conference DE-SIGN 2010* (Editors: Marjanovic et al.), pages 49–58.
[JS]   Jones, A.; Subrahmanian, E. (2013) "A Consilience-based Approach to Engineering Services in Global Supply Chains". *Proceedings of the Advanced Production Management Systems Conference*, September 9-12, 2013, State College, Pennsylvania. *To appear.*
[Lei]   Leinster, T. (2004) *Higher Operads, Higher Categories.* London Mathematical Society Lecture Note Series 298, Cambridge University Press.
[Lur]   Lurie, J. (2012) "Higher algebra". http://www.math.harvard.edu/~lurie/papers/HigherAlgebra.pdf.
[Mac]   (1998) S. Mac Lane. *Categories for the working mathematician.* Second edition. Graduate Texts in Mathematics, 5. Springer-Verlag, New York.
[Man]   Manzyuk, O. (2009) "Closed categories vs. closed multicategories". http://arxiv.org/abs/0904.3137
[May]   May, P. (1972). *The geometry of iterated loop spaces.* Springer-Verlag.
[Min]   Minsky, M. (1988). *The society of mind.* Simon and Schuster.
[Pie]   Pierce, B. (1991) *Basic category theory for computer scientists.* MIT Press.
[RS]   Radul, A.; Sussman, G.J. (2009). "The art of the propagator". *MIT Computer science and artificial intelligence laboratory technical report.*
[Sp1]   Spivak, D.I. (2009) "Simplicial databases". http://arxiv.org/abs/0904.2012
[Sp2]   Spivak, D.I. (2010) "Table manipulation in simplicial databases". http://arxiv.org/abs/1003.2682
[Sp3]   Spivak, D.I. (2012) "Functorial data migration". *Information and communication.*
[Sp4]   Spivak, D.I. (2013) *Category theory for scientists.* http://arxiv.org/abs/1302.6946
[SS]   Spivak, D.I.; Stapleton, N. "The directed wiring diagram operad and its algebra of functions," *In preparation.*

Department of Mathematics, Massachusetts Institute of Technology, Cambridge MA 02139

E-mail address: dspivak@math.mit.edu