NOTES ON MULTIGRAPHS

IORDAN GANEV

Version 1.0

CONTENTS

1. INTRODUCTION

These expository notes collect basic constructions and results related to multigraphs, multicategories, and related structures. In this introductory section, we motivate and illustrate elementary versions of the key ideas.

1.1. **Graphs as generalized functions.** We begin with the observation that a function $f : X \to X$ from a set X to itself defines a directed graph: the vertex set is X and there is a directed edge from x to $f(x)$ for every $x \in X$. Thus, the edge set is the "graph of f":

$$\mathrm{Graph}(f) = \{(x, f(x)) \in X \times X \mid x \in X\} \to X \times X$$

with the projections onto the two factors giving the source and target maps. Hence, we can regard directed graphs as generalizations of functions. Specifically, suppose

$$E \to X \times X$$

is a directed graph on the vertex set X, where E is the set of (directed) edges, and the maps to the factors indicate the source and target of each edge. Given a second directed graph $E' \to X \times X$, we can define the composition of E and E' as the fiber product:

$$E \times_X E' = \{(e, e') \in E \times E' \mid t(e) = s(e')\} \tag{1.1}$$

so that (e, e') belongs to the composition if and only if the target of e is the source of e'. The composition is again a directed graph on the vertex set X, where the source and target of (e, e') are, respectively, the source of e and the target of e'. In particular, for functions $f : X \to X$ and $g : X \to X$, composition of directed graphs reflects composition of functions:

$$\text{Graph}(f) \times_X \text{Graph}(g) = \text{Graph}(g \circ f)$$

The composition in 1.1 is easily seen to be associative and unital, with the unit given by the diagonal map $X \to X \times X$, namely, the directed graph with a loop at each vertex and no other edges; it is also the graph of the identity function $\text{id} : X \to X$.

1.2. **Spans.** What about functions whose domain and codomain differ? If $f : X \to Y$ is a function of sets, then its graph lies over $X \times Y$ rather than over $X \times X$:

$$\text{Graph}(f) = \{(x, f(x)) \in X \times Y \mid x \in X\} \to X \times Y$$

Recall that a *span* from X to Y is a set E with maps:

$$\begin{tikzcd} & E \arrow[dl] \arrow[dr] & \\ X & & Y \end{tikzcd}$$

So each function $f : X \to Y$ defines a span from X to Y via its graph, and a directed graph is the same thing as an *endospan*, i.e., a span from a set to itself. Observe that spans from X to Y are nothing more than objects of the slice category in Set over $X \times Y$, and we can write a span more succinctly as $E \to X \times Y$; we move freely between these two perspectives. The composition rule for directed graphs extends easily to spans: given $(s, t) : E \to X \times Y$ and $(s', t') : E' : Y \to Z$, we form the fiber product:

$$E \times_Y E' = \{(e, e') \in E \times E' \mid t(e) = s'(e')\} \xrightarrow{s \times t'} X \times Z \tag{1.2}$$

Thus we have a map:

$$\begin{aligned} \mathsf{Span}(X, Y) \times \mathsf{Span}(Y, Z) &\longrightarrow \mathsf{Span}(X, Z) \\ E, E' &\longmapsto E \times_Y E' \end{aligned}$$

This composition is associative and unital, with the diagonals $X \to X \times X$ playing the role of unit. Moreover, there is a category of spans where the objects are sets and where the morphisms from X to Y are given by spans from X to Y. Assigning an ordinary function to its graph defines a functor from the category of sets to the category of spans.

1.3. **Lists and composition.** Regarding directed graphs as functions will be useful when considering functions involving lists; a point we will return to shortly. For now, as a warm-up, suppose X, Y, and Z are sets and we have functions:

$$f : X \to \mathsf{List}(Y), \qquad g : Y \to \mathsf{List}(Z)$$

where $\mathsf{List}(-)$ denotes the set of finite sequences of elements in a set. Although the codomain of f is not precisely equal to the domain of g, concatenation of lists allows us to define a composition rule for f and g, namely:

$$X \xrightarrow{f} \mathsf{List}(Y) \xrightarrow{\mathsf{List}(g)} \mathsf{List}(\mathsf{List}(Z)) \xrightarrow{\text{concat}} \mathsf{List}(Z) \tag{1.3}$$

Hence we have a composition map, known as *Kliesli composition*[1]:

$$\begin{aligned} \circ_{\mathsf{List}} : \mathsf{Set}(Y, \mathsf{List}(Z)) \times \mathsf{Set}(X, \mathsf{List}(Y)) &\longrightarrow \mathsf{Set}(X, \mathsf{List}(Z)) \\ g, f &\longmapsto \mathsf{concat} \circ \mathsf{List}(g) \circ f \end{aligned}$$

where $\mathsf{Set}(-,-)$ denotes functions between sets, and $\mathsf{List}(g)$ is defined as applying g to every element in the input list. We observe:

- The composition is associative in the sense that, if we have $h : Z \to \mathsf{List}(W)$, then

  $$h \circ_{\mathsf{List}} (g \circ_{\mathsf{List}} f) = (h \circ_{\mathsf{List}} g) \circ_{\mathsf{List}} f.$$

  This equation follows easily from the associativity of list concatenation.

- The composition is unital in the following sense. Let $\epsilon_X : X \to \mathsf{List}(X)$ be the map sending x to the singleton list $[x]$, and similarly define ϵ_Y. Then, for $f : X \to \mathsf{List}(Y)$, we have $f \circ_{\mathsf{List}} \epsilon_X = f$ and $\epsilon_Y \circ_{\mathsf{List}} f = f$ as maps $X \to \mathsf{List}(Y)$.

Now we turn things around. Suppose we have maps:

$$f : \mathsf{List}(X) \to Y \qquad g : \mathsf{List}(Y) \to Z$$

so that we have placed lists in the domain rather that the codomain. Is there a composition rule for these maps? In other words, we seek to produce a function $\mathsf{List}(X) \to Z$ from f and g in a way that satisfies associativity and unit axioms. The first attempt would be to form the composition:

$$\mathsf{List}(\mathsf{List}(X)) \xrightarrow{\mathsf{List}(f)} \mathsf{List}(Y) \xrightarrow{g} Z$$

and precompose with a map from $\mathsf{List}(X)$ to $\mathsf{List}(\mathsf{List}(X))$. What would this map be? There are two candidates, namely $\epsilon_{\mathsf{List}(X)}$ and $\mathsf{List}(\epsilon_X)$. We invite the reader to verify that neither choice leads to satisfactory associativity and unit axioms for composition. In essence, both maps loose too much information.

1.4. List-Spans.

A way out of this difficulty is to return to spans as an expanded notion of functions, but with a twist to accommodate for lists. Specifically, define a List-*span* from X to Y to be span from $\mathsf{List}(X)$ to Y; that is, a set E with maps:

$$\begin{tikzcd} & E \arrow[dl] \arrow[dr] & \\ \mathsf{List}(X) & & Y \end{tikzcd}$$

An ordinary function $\mathsf{List}(X) \to Y$ becomes a List-span by taking E to be the graph of f. We define the following composition rule for List-spans. Given $E \to \mathsf{List}(X) \times Y$ and

---

[1] This is the composition law in the Kliesli category for List, where the objects are sets and morphisms from X to Y are maps of sets from X to $\mathsf{List}(Y)$.

$E' \to \mathsf{List}(Y) \times Z$, we form the fiber product:

$$\begin{tikzcd}
 & & & \mathsf{List}(E) \times_{\mathsf{List}(Y)} E' \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\wedge"] & & \\
 & & \mathsf{List}(E) \arrow[dl] \arrow[dr] & & E' \arrow[dl] \arrow[dr] & \\
\mathsf{List}(X) & \mathsf{List}(\mathsf{List}(X)) \arrow[l, "\mathrm{concat}"'] & & \mathsf{List}(Y) & & Z
\end{tikzcd}$$

where, in the left portion of the diagram, we apply the functor List to the map $E \to \mathsf{List}(X) \times Y$, and concatenate the resulting lists of lists of X. The diagram indicates that $\mathsf{List}(E) \times_{\mathsf{List}(Y)} E'$ is a span from $\mathsf{List}(X)$ to Z, as desired. Thus we have a map:

$$\begin{aligned}
\mathsf{Span}(\mathsf{List}(X), Y) \times \mathsf{Span}(\mathsf{List}(Y), Z) &\longrightarrow \mathsf{Span}(\mathsf{List}(X), Z) \\
E, E' &\longmapsto \mathsf{List}(E) \times_{\mathsf{List}(Y)} E'
\end{aligned}$$

We comment on the unit and associativity axioms:

- For any set X, consider the List-span $X \to \mathsf{List}(X) \times X$ taking x to $([x], x)$. This satisfies the left and right unit axioms for the composition:

  $$\mathsf{List}(X) \times_{\mathsf{List}(X)} E = E \qquad \mathsf{List}(E) \times_{\mathsf{List}(X)} X = E$$

  The first identity is immediate; the second requires a short argument.

- Suppose, in addition to $E_1 = E$ and $E_2 = E'$ from above, we have $E_3 : \mathsf{List}(Z) \to W$. There is a natural map:

  $$\mathsf{List}(\mathsf{List}(E_1) \times_{\mathsf{List}(Y)} E_2) \times_{\mathsf{List}(Z)} E_3 \longrightarrow \mathsf{List}(E_1) \times_{\mathsf{List}(Y)} \mathsf{List}(E_2) \times_{\mathsf{List}(Z)} E_3$$

  given by applying $\mathrm{concat} \circ \mathsf{List}(\pi_1)$ and $\mathsf{List}(\pi_2)$, where π_1, π_2 are the projections from $\mathsf{List}(E_1) \times_{\mathsf{List}(Y)} E_2$ to each of the factors, and applying the identity map to E_3. One argues that this map is an isomorphism (the argument is slightly technical but not difficult).

We have thus succeeded in defining a notion of composition of functions with List in the domain, at the cost of using a broader definition of functions.

This concludes the elementary prelude to these notes. Below, we will see how composition law for List-spans can be used to define a *multicategory*. We will be most interested in free multicategories, which are those generated by List-endospans, otherwise known as *multigraphs*. Finally, one can consider replacing the monad List with a general monad T; depending on the properties of the monad, one can define a composition law for T-spans.

1.5. **Summary of these notes.** We now summarize the contents of these notes. In Section 2 we first briefly recall facts about directed graphs and paths therein, and then formally introduce multigraphs. Multigraphs differ from ordinary directed graphs in that edges can have a list of multiple source vertices (with multiplicities) while retaining a single target vertex.

As mentioned above, the composition of List-spans described in Section 1.4 is a special case of composition in a multicategory. We discuss multicategories in Section 3, first introducing the notion in general and then examining several equivalent perspectives on the free multicategory generated by a multigraph. One perpsective is based on trees, and the other based on initial algebras. We list a number examples of multigraphs and the structure captured by terms in the associated free multicategories. Finally, we discuss positional encodings of multimorphisms.

In Section 4 we move beyond the List endofunctor to consider T-spans for a cartesian monad T. These concepts lead to the construction of a bicategory $\mathsf{Set}_{(T)}$ that is a higher version of the category of spans noted in Section 1.4 above. A T-multicategory is a monad in the bicategory $\mathsf{Set}_{(T)}$, and generalizes the notion of plain multicategories (which can be recovered in the case of $T = \mathsf{List}$). We examine the notion of free T-multicategories in detail, with several running examples, before concluding with comments on the composition of cartesian monads.

In Section 5 we consider the Bag endofunctor, which is not cartesian. There is no associated T-multicategory, but there is still a notion of ”quasimultimorphism”, which admits an interpretation in terms of tree. There is a quotient map from multimorphisms to quasimultimorphisms, and we also have a notion of positional encodings for quasimultimorphisms.

Finally, in Section 6, we collect information about various constructions carrying the adjective ”polynomial”, including polynomial spans and polynomial functors. We explain the relationship with usual polynomials, and with mulitgraphs.

Appendix A records further observations about the relation between multimorphisms and quasimultimorphisms. Appendix B comprises a proof of associativity for free T-multicategories.

1.6. **Notation.** We denote the set of natural numbers as $\mathbb{N} = \{0, 1, 2, \dots\}$. For $n \in \mathbb{N}$, we set $[n]$ to be the ordered set $\{1, 2, \dots, n\}$, with the convention that $[0] = \emptyset$.

We denote the category of sets as Set. We write $X + Y$ for the coproduct (disjoint union) of sets. Given maps $f : X \to Y$ and $g : Y \to Z$, we write (f, g) for the induced map $X + Y \to Z$. We also write $\sum_i X_i$ for the coproduct of an indexed family of sets. Given a set X, the *slice category over* X has objects given by maps of sets $E \to X$, and a morphism from $E_1 \to X$ to $E_2 \to X$ is a map $f : E_1 \to E_2$ intertwining the maps to X.

The endofunctor $\mathsf{List} : \mathsf{Set} \to \mathsf{Set}$ takes a set X to the set of finite sequences of elements of X, that is, $\mathsf{List}(X) = \sum_{n \in \mathbb{N}} X^n$. An element of $\mathsf{List}(X)$ is either the empty list, denoted $[]$, or non-empty and written as $\mathbf{x} = [x_1, \dots, x_n]$, with each $x_i \in X$. If $f : X \to Y$, we have an induced map $\mathsf{List}(f) : \mathsf{List}(X) \to \mathsf{List}(Y)$ that applies f to each list element, and preserves the length. The List endofunctor is a monad with multiplication $\mu : \mathsf{List} \circ \mathsf{List} \to \mathsf{List}$ and unit $\epsilon : \mathrm{id} \to \mathsf{List}$ given by:

$$\mu_X : \mathsf{List}(\mathsf{List}(X)) \to \mathsf{List}(X) \qquad\qquad \epsilon_X : X \to \mathsf{List}(X)$$
$$[\mathbf{x}_1, \dots, \mathbf{x}_n] \mapsto \mathbf{x}_1 + \cdots + \mathbf{x}_n \qquad\qquad x \mapsto [x]$$

where summation is concatenation, noting that concatenation is a non-commutative operation. (The use of summation symbols is overloaded in this document; their meaning will be clear from context.)

The endofunctor $\mathsf{Bag} : \mathsf{Set} \to \mathsf{Set}$ takes a set X to the set of functions $X \to \mathbb{N}$ with finite support. It has a monad structure $\mu_X : \mathsf{Bag}(\mathsf{Bag}(X)) \to \mathsf{Bag}(X)$ sending $\beta : \mathsf{Bag}(X) \to \mathbb{N}$ to the bag $x \mapsto \sum_{b \in \mathsf{Bag}(X)} \beta(b) b(x)$. The unit map picks out delta functions. We may also write a bag $b : X \to \mathbb{N}$ as a formal sum $\sum_x b(x)x$.

We briefly recall the essential components of the definition of a bicategory; see [Lei04, Definition 1.5.1] for the full definition. A bicategory consists of a collection of objects (called 0-cells), and, for every pair of 0-cells there is a category $\mathcal{B}(V, W)$, whose objects are called 1-cells and whose morphisms are called 2-cells. Additionally, there is a composition functor $\mathcal{B}(W, Z) \times \mathcal{B}(V, W) \to \mathcal{B}(V, Z)$, an identity object $1_V \in \mathcal{B}(V, V)$ for each 0-cell V, associativity coherence isomorphisms, and unit coherence isomorphisms.

2. Graphs and multigraphs

We begin with a recollection of the definition of a directed graph, and paths in a directed graph; we subsequently introduce multigraphs.

2.1. **Graphs.** A (directed) graph Γ consists of a set V of vertices and a set E of edges, together with a map:

$$E \to V \times V$$

indicating the source vertex and target vertex of each edge. Hence, a graph on a vertex set V is the same thing as an object of the slice category in Set over $V \times V$. A path in the graph Γ is either:

1. a constant path at a vertex $v \in V$, or
2. a non-empty list $p = [e_1, \dots, e_n]$ of edges such that the target of e_i is equal to the source of e_{i+1}.

In the latter case, the source and target of p are the source of e_1 and the target of e_n, respectively. We denote by $\mathrm{Paths}(\Gamma)$ the set of all paths in the graph Γ. Thinking of edges as length-one paths, the map $E \to V \times V$ from above extends to a map:

$$\mathrm{Paths}(\Gamma) \to V \times V$$

taking a path from v to w to the pair (v, w). Path concatenation is a map:

$$\mathrm{Paths}(\Gamma) \times_V \mathrm{Paths}(\Gamma) \to \mathrm{Paths}(\Gamma)$$

where, in forming the fiber product, the map to V from the first (resp. second) factor is the target (resp. source) map. Path concatenation is associative, and the constant paths act as units. Paths are graded by their length, and concatenation is additive with respect to this grading.

2.2. **Multigraphs.** Multigraphs differ from graphs in that we allow each edge to have a list of source vertices, while retaining a single target. In more detail:

Definition 2.1. A *multigraph* consists of a set V of vertices and a set E of *multi-edges*, or just *edges* for short, together with a map:

$$E \to \mathsf{List}(V) \times V$$

indicating the list of source vertices and the single target vertex of each edge[2]. The *arity* of an edge is the length of its source list, and an n-ary edge is one with source list of length n. If $n = 0$, 1, or 2, we say that the edge is *nullary*, *unary*, or *binary*, respectively.

Hence, a multigraph on vertex set V is the same thing as an object of the slice category in Set over $\mathsf{List}(V) \times V$. One way to depict an edge of a multigraph is as an upward pointing triangle with its target vertex at the apex, and its sources listed below. We refer to this convention as *constructor notation*. Three examples appear in Figure 1. On the left, we have a multigraph with a single vertex v, a nullary edge, and a binary edge. In the middle, we have a multigraph with vertex set $\{u, w\}$, a nullary edge with target u, a unary edge with target w and source u, and a unary edge with target u and source w. On the right, we have a multigraph with two vertices l and a, a nullary edge with target l, and a binary edge with target l and sources a and l.

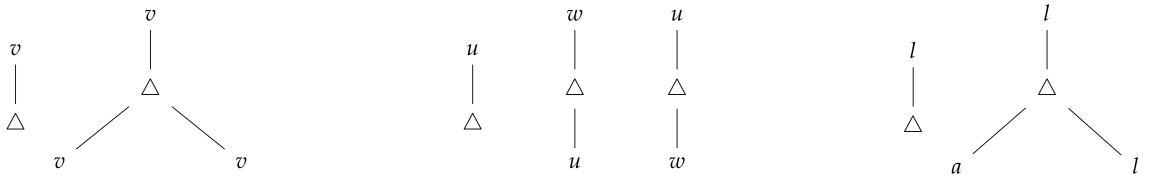

Figure 1. Three multigraphs depicted using constructor notation.

An alternative way to depict an edge is using a horizontal bar, where the sources of the arguments are listed above the bar, and the target vertex is at the bottom. We refer to this as *inference line notation* (Figure 2). While constructor notation is generally read from bottom up, inference line notation is generally read from the top down.

$$\frac{}{v} \quad \frac{v \quad v}{v} \qquad\qquad \frac{}{u} \quad \frac{w}{u} \quad \frac{u}{w} \qquad\qquad \frac{}{l} \quad \frac{a \quad l}{l}$$

Figure 2. Three multigraphs depicted using inference line notation.

Further examples appear in Section 3.4 below, once we have introduced the notion of *terms* of a mulitgraph.

---

[2] The vertices of a multigraph may also be referred to as *sorts* or *types*, and the multi-edges as *constructors*.

3. MULTICATEGORIES

A graph gives rise to a free category whose objects are the set of vertices, and whose morphisms are directed paths through the graph. Analogously, a multigraph gives rise to a free *multicategory*. We give an expository account of multicategories in general before exploring the free multicategory generated by a multigraph in more detail.

3.1. **Multicategories in general.** Just like an ordinary (small) category, a *multicategory* has a set Obj of objects; however, rather than having a set of morphisms only between any pair of objects, a multicategory has a set of *multimorphisms* $\mathsf{MMor}(\mathbf{a}, b)$ for any pair consisting of a list $\mathbf{a} = [a_1, \ldots, a_n]$ of source objects and a single target object b. The composition is given by a map:

$$[\mathsf{MMor}\left(\mathbf{a}_1, b_1\right), \ldots, \mathsf{MMor}\left(\mathbf{a}_n, b_n\right)] \times \mathsf{MMor}\left(\mathbf{b}, c\right) \to \mathsf{MMor}\left(\mathbf{a}_1 + \cdots + \mathbf{a}_n, c\right)$$

where $\mathbf{b} = [b_1, \ldots, b_n]$ and $\mathbf{a}_1 + \cdots + \mathbf{a}_n$ denotes the concatenation of the lists $\mathbf{a}_1$ through $\mathbf{a}_m$, in order. In addition, we have an identity element in $\mathsf{MMor}([a], a)$ for any object a. The composition map is required to satisfy the appropriate unit and associativity axioms Slightly more formally:

Definition 3.1. A multicategory $\mathcal{C}$ consists of:

(1) a set Obj of objects
(2) a set MMor of multimorphisms with a map

$$(s, t) : \mathsf{MMor} \to \mathsf{List}(\mathsf{Obj}) \times \mathsf{Obj}$$

(3) a composition map

$$\mathsf{List}(\mathsf{MMor}) \times_{\mathsf{List}(\mathsf{Obj})} \mathsf{MMor} \to \mathsf{MMor}$$

where the fiber product is:

$$\{([f_1, \ldots, f_n], f) \in \mathsf{List}(\mathsf{MMor}) \times \mathsf{MMor} \mid [t(f_1), \ldots, t(f_n)] = s(f)\}$$

with $(s, t) \circ \mathsf{compose} = (\mathsf{concat}(\mathsf{List}(s) \circ \pi_1)\ ,\ t \circ \pi_2)$
(4) a unit map $\eta : \mathsf{Obj} \to \mathsf{MMor}$ with $(s, t) \circ \eta(v) = ([v], v)$.

satisfying associativity and unital axioms that we do not make explicit here but will do so in a more abstract setting in Section 4.3 below (see also [Lei04]). The *terms* of a multicategory is the set of multimorphisms with empty list of sources:

$$\mathsf{Terms}_{\mathcal{C}} = s^{-1}([\,]) = \{f \in \mathsf{MMor} \mid s(f) = [\,]\}$$

Similarly, the *terms* with target v, denoted $\mathsf{Terms}(v)$, are those terms f with $t(f) = v$.

The main examples of multicategories relevant to this document are free multicategories generated by a multigraph, which we discuss below. For now, we include a different sort of example:

Example 3.2. There is a multicategory of vector spaces with multilinear maps, where the objects are vector spaces over a fixed field k and $\mathsf{MMor}([V_1, \ldots, V_n], V)$ is the set of multilinear maps $V_1 \times \cdots \times V_n \to V$. This operation can be captured (or obscured, depending one's inclination) by the tensor product construction:

$$\mathsf{MMor}([V_1, \ldots, V_n], V) = \mathrm{Hom}_k(V_1 \otimes \cdots \otimes V_n, V)$$

where Hom_k indicates linear maps. The set of terms with target V is $\mathrm{Hom}_k(k, V) = V$. This construction can be adapted to provide a multicategory corresponding to any monoidal category.

Example 3.3. Let M be a commutative monoid. There is a multicategory with one object, call it $\bullet$, and multimorphisms given by $\mathsf{MMor} = M \times \mathbb{N}$, where the map to $\mathsf{List}(\bullet) \times \bullet = \mathbb{N}$ is the projection. Thus,

$$\mathsf{MMor}([\bullet]^n, \bullet) = M$$

for every n, with $[\bullet]^n$ abbreviating the list of length n. Composition uses the monoidal operation, the unit in $\mathsf{MMor}([\bullet], \bullet) = M$ is the unit of M, and associativity follows uses the commutativity of M.

3.2. **The free multicategory: trees.** Returning to multigraphs, there is a *free multicategory* generated by a multigraph Γ. Its objects are the vertices of the multigraph and its multimorphisms admit two closely related descriptions. We start with a somewhat informal description of multimorphisms as rooted trees in which every node is a copy of a edge (think of the illustration in Figure 1), and gluing compatibilities are determined by matching target and source. More precisely:

- There is an identity multimorphism in $\mathrm{Hom}([v], v)$ for any vertex $v \in V$. These are depth zero trees.
- There is a multimorphism for every edge e, and it belongs to $\mathrm{Hom}(s(e), t(e))$. These are depth one trees.
- To obtain a multimorphism of depth two, one starts with a non-nullary edge e, and attaches copies of edges to some of the vertices in the source list of e so that the target of the attached edge matches the source it is attached to. The unmatched sources, from either the original edge or the attached ones, form the list of inputs for the multimorphism.
- Then one attaches more copies of edges to some of the added edges, and so on for finitely many steps.

Multimorphisms in a free multicategory are filtered by depth: a multimorphism of depth d composed with n multimorphisms of depths $d_i, \ldots, d_n$ has depth at most $d + \max(d_i)$. In symbols:

$$\mathrm{depth}\left(\mathsf{compose}\left([f_1, \ldots, f_n], g\right)\right) \leq \max_i \left(\mathrm{depth}\left(f_i\right)\right) + \mathrm{depth}(g)$$

One can also speak about the depth of a particular node in the tree. At any stage there is a list of "dangling gaps" at the bottom of the tree; these are labeled by vertices and form the source list of the multimorphism, where further multimorphisms (trees) can be attached.

3.3. **The free multicategory: initial algebras.** Equivalently, one can define the set of multimorphisms in the free multicategory generated by a multigraph Γ as the initial algebra of the endofunctor F_Γ on the slice category $\mathsf{Set}/\left(\mathsf{List}(V) \times V\right)$ defined as follows. Given an object $\alpha : X \to \mathsf{List}(V) \times V$ of the slice category, consider the fiber product at the top of the following diagram:

$$\begin{tikzcd}
 & & & \mathsf{List}(X) \times_{\mathsf{List}(V)} E \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] \arrow[d, phantom, "\wedge"] & & \\
 & & \mathsf{List}(X) \arrow[dl, "\mathsf{List}(\pi_2 \circ \alpha)"'] \arrow[dr, "\mathsf{List}(\pi_1 \circ \alpha)"] & {} & E \arrow[dl, "s"'] \arrow[dr, "t"] & \\
\mathsf{List}(V) & \mathsf{List}(\mathsf{List}(V)) \arrow[l, "\mu"] & & \mathsf{List}(V) & & V
\end{tikzcd}$$

The diagram indicates that the fiber product is naturally an object of the slice category over $\mathsf{List}(V) \times V$. Additionally, the vertex set V is an object of the slice category via

$$V \to \mathsf{List}(V) \times V, \qquad v \mapsto ([v], v)$$

We define an endofunctor of the slice category over $\mathsf{List}(V) \times V$ as:

$$F_\Gamma : X \mapsto V + \mathsf{List}(X) \times_{\mathsf{List}(V)} E$$

Lemma 3.4. *The functor F_Γ has an initial algebra.*

Proof. It suffices to show that $F = F_\Gamma$ is finitary, that is, that every element of $F(X)$ lies in the image of $F(Y)$ for some finite subset $Y \subseteq X$ [AMMU15]. To this end, take $z \in F(X)$. If $z \in V$, we take $Y = \emptyset$. Otherwise, $z = (\mathbf{x}, e)$ for some list $\mathbf{x} = [x_1, \ldots, x_n]$ of elements of X, and some edge e such that $s(e) = [\pi_1 \circ \alpha(x_1), \ldots, \pi_1 \circ \alpha(x_n)]$. Take $Y = \{x_1, \ldots, x_n\}$. $\square$

Definition 3.5. Let $\Gamma : E \to V \times V$ be a multigraph. Define a multicategory with:

(1) Objects are the vertices: $\mathsf{Obj} = V$.
(2) Multimorphisms are the initial algebra of the endofunctor F_Γ

We define the set of multimorphisms $\mathsf{MMor}_\Gamma = \mathsf{MMor}$ as the initial algebra of this endofunctor, which can be obtained inductively as the union of filtered pieces:

$$\mathsf{MMor}^{(0)} = V$$

$$\mathsf{MMor}^{(n)} = F_\Gamma\left(\mathsf{MMor}^{(n-1)}\right) = V + \mathsf{List}\left(\mathsf{MMor}^{(n-1)}\right) \times_{\mathsf{List}(V)} E \qquad \text{for } n \geq 1$$

The fixed point property of the initial algebra gives the glueing map:

$$\mathsf{glue} : V + \mathsf{List}(\mathsf{MMor}) \times_{\mathsf{List}(V)} E \xrightarrow{\sim} \mathsf{MMor}$$

Since the glueing map is an isomorphism, we also have a "deglueing" map.

$$\mathsf{deglue} : \mathsf{MMor} \xrightarrow{\sim} V + \mathsf{List}(\mathsf{MMor}) \times_{\mathsf{List}(V)} E$$

The deglueing map indicates that a non-identity multimorphism has a unique "head" edge and a unique list of "children" multimorphisms attaching to that edge. The glueing map can be iterated to obtain the composition map of multimorphisms:

$$\mathsf{compose} : \mathsf{List}(\mathsf{MMor}) \times_{\mathsf{List}(V)} \mathsf{MMor} \longrightarrow \mathsf{MMor}$$

which is surjective, but not injective. The formulation of the associativity and unital axioms is slightly involved, and their proof is technical; for this reason we do not make them explicit here, but they follow from results in Section 4.3 and Appendix B.

Before moving on to examples, we connect with the initial algebra description of multimorphisms with the tree perspective. The filtered piece $\mathsf{MMor}^{(d)}$ corresponds to trees of depth at most d. The composition map restricts to:

$$\mathsf{List}(\mathsf{MMor}^{(d)}) \times_{\mathsf{List}(V)} \mathsf{MMor}^{(d')} \longrightarrow \mathsf{MMor}^{(d+d')}$$

In other words, composing a tree of depth at most d with a list of compatible trees of depth at most d' leads to a tree of depth at most $d + d'$. A more refined version is:

$$\left(\mathsf{MMor}^{(d_1)} \times \cdots \times \mathsf{MMor}^{(d_k)}\right) \times_{V^k} \mathsf{MMor}^{(d')} \longrightarrow \mathsf{MMor}^{(\max_i(d_i)+d')}$$

which is a restatement of the observation above that attaching trees of depth at most $d_1, \ldots, d_k$ to a tree of depth at most d results in a tree of depth at most $\max_i(d_i) + d$.

3.4. **Examples.** The examples below are generally named based on a common description of the terms of the corresponding free multicategory; unless specified otherwise, there is only one sort. For brevity, we write "terms of the multigraph" instead of "terms of the free multicategory generated by the multigraph".

(1) The set of natural numbers is the set of terms of the multigraph with a single nullary zero and a single unary succ.

(2) The set of binary trees is the set of terms of the multigraph with a single nullary leaf and a single binary node. See the left example in Figure 1.

(3) For $k > 0$, the set of k-nary trees is the set of terms of the multigraph with a single nullary leaf and a single k-nary constructor k-node. We recover the natural numbers and the set of binary trees when $k = 1$ and $k = 2$, respectively.

(4) The set of rose trees is the set of terms of the multigraph with a single n-ary constructor for $n = 0, 1, \ldots$. Thus, the number of constructors is countably infinite.

(5) For $k > 0$, the set of k-lily trees is the set of terms of the multigraph with a single n-ary constructor for every $n = 0, 1, \ldots, k$. We recover the natural numbers when $k = 1$. So lily trees are a truncation of rose trees.

(6) Consider the second multigraph in Figure 1. There are two sorts, called u and w. There is a single nullary constructor zero with target u and two unary constructors, one from $[u]$ to w and one from $[w]$ to u. The terms with target u are even numbers, while the terms with target w are odd numbers.

(7) Consider the third multigraph in Figure 1. As stated, there is only one term, namely that corresponding to the nullary edge. However, suppose that, for a given set X, we introduce a nullary edge with target a for each element $x \in X$. Then the terms are lists of elements of X, where the nullary edge corresponds to the empty list.

(8) Similarly, we form k-nary trees and lily trees with elements of a set X by adding an extra sort X, nullaries with target X for each element of X, and a source sort X for every previous constructor.

(9) The set of propositional formulas on variables X is the set of terms of the multigraph with a nullary for every $x \in X$, a nullary corresponding to $\top$ (True), a nullary corresponding to $\perp$ (False), a unary $\neg$ (not), and binaries for each of $\Rightarrow$ (implies), $\wedge$ (and), $\vee$ (or).

(10) For $k > 0$, recall that the set of Dyck-k words is the set of balanced expressions of k paris of delimiters (i.e., k types of open-closed bracket pairs). This coincides with the set of terms of the multigraph with a single nullary and k binary constructors.

(11) For a set X, the set of simply-typed lambda terms over X is the set of terms of the multigraph with a nullary for every $x \in X$, a unary abstraction constructor, and a binary application constructor.

3.5. **Traversals and encodings.** We now discuss various ways to "flatten" a multimorphism (aka tree) into a list. We begin with the preorder traversal of a multimorphism, which is defined as:

$$\begin{aligned} \text{PostOrder} : \mathsf{MMor}_\Gamma &\to \mathsf{List}(E) \\ \mathrm{id}_v &\mapsto [] \\ \mathsf{glue}\left([f_1, f_2 \dots, f_n], e\right) &\mapsto \left(\sum_{i=1}^{n} \text{PostOrder}(f_i)\right) + [e] \end{aligned}$$

where summation in the last line denote concatenation of lists, in order. Hence, the postorder traversal flattens a multimorphism into a list by performing a depth-first search[3]. The *length* or *number of nodes* in a multimorphism is the length of its image in $\mathsf{List}(E)$ under either the above traversals.

The postorder traversal is not injective, and so one cannot recover a multimorphism from its preorder or postorder traversal. However, one can recover any (non-constant) multimorphism from its set of nodes, appropriately defined. Recall that that a node in a (rooted) tree is determined by its path to the root, where the path consists of the sequence of edges needed to arrive at the root, with extra data specifying, for each edge, which input is being used. To make this precise, we introduce the set of arguments of a multigraph:

---

[3] The preorder traversal map is defined in the same way, except $[e]$ is concatenated at the beginning.

Definition 3.6. Let $E \to \mathsf{List}(V) \times V$ be a multigraph.

(1) The set of *arguments* is defined as:

$$A = \left(\sum_{n \in \mathbb{N}} [n] \right) \times_{\mathbb{N}} E = \{(i, e) \in \mathbb{N} \times E \mid 1 \leq i \leq \mathsf{arity}(e)\}$$

(2) Define $h : A \to V$ by taking (i, e) to $s(e)[i]$, that is, the i-th source vertex of the source list of the edge e.

(3) Define $g : A \to V$ as the projection map $(i, e) \mapsto e$.

(4) A *node* in a multimorphism is a nonempty list in $\mathsf{List}(A + E)$ of the form:

$$[(i_1, e_1), e_1, \dots, (i_n, e_n), e_n] \qquad \text{or} \qquad [e_0, (i_1, e_1), e_1, \dots, (i_n, e_n), e_n]$$

where $t(e_j) = s(e_{j+1})[i_{j+1}]$.

(5) To any multimorphism we define its *list of nodes* via:

$$\begin{aligned} \mathrm{Nodes} : \mathsf{MMor}_\Gamma &\to \mathsf{List}\left(\mathsf{List}\left(A + E\right)\right) \\ \mathrm{id}_v &\mapsto [] \\ \mathsf{glue}\left([f_1, f_2 \dots, f_n], e\right) &\mapsto \left(\sum_{i=1}^{n} \sum_{j=1}^{m_i} \left[\gamma_j^{(i)} + [(i, e), e]\right] \right) + [[e]] \end{aligned}$$

where $\mathrm{Nodes}(f_i) = \left[\gamma_1^{(i)}, \dots, \gamma_{m_i}^{(i)}\right] \in \mathsf{List}\left(\mathsf{List}\left(A + E\right)\right)$.

In other words, we identify a node with its path to the root; moreover, in the inductive step for defining the list of paths, to each path $\gamma_j^{(i)}$, we append the elements $(i, e) \in A$ and $e \in E$. One argues by induction that the above map of taking nodes is injective on non-constant multimorphisms.

3.6. **Positional encodings.** In order to obtain positional encodings, we go one step further to produce a list of vectors.

Definition 3.7. A d-dimensional positional encoding representation is a map:

$$\rho : A + E \to \mathbb{R}^{d \times d}$$

The corresponding d-dimensional encoded (preorder) traversal is the map:

$$\begin{aligned} \Psi : \mathsf{MMor}_\Gamma \times \mathbb{R}^d &\to \mathsf{List}\left(\mathbb{R}^d\right) \\ (\mathrm{id}_v, u) &\mapsto [] \\ \left(\mathsf{glue}\left([f_1, f_2 \dots, f_n], e\right)\ ,\ u\right) &\mapsto [M_e u] + \sum_{i=1}^{n} \Psi\left(f_i\ ,\ M_{(i,e)} M_e u\right) \end{aligned}$$

where summation denotes concatenation of lists in order, and $M_{(i,e)}$ and M_e denote the matrices $\rho((i, e))$ and $\rho(e)$, respectively.

The following diagram commutes:

$$\begin{tikzcd}
\mathsf{List}(\mathsf{List}(A+E)) \arrow[r, "\mathsf{List}(\mathsf{List}(\rho))"] & \mathsf{List}(\mathsf{List}(\mathbb{R}^{d\times d})) \arrow[r, "\mathsf{List}(\mathsf{mult})"] & \mathsf{List}(\mathbb{R}^{d\times d}) \arrow[d, "\mathsf{List}(\mathrm{ev}_u)"] \\
\mathsf{MMor}_\Gamma \arrow[u, "\mathrm{Nodes}"] \arrow[rr, "\Psi(-,u)"'] & & \mathsf{List}(\mathbb{R}^d)
\end{tikzcd}$$

where the map mult multiplies a list of d by d matrices to produce a single d by d matrix, the map ev_u multiplies a d by d matrix by the d-vector u to produce another d vector, and the map $\Psi(-,u)$ sends a multimorphism f to $\Psi(f,u)$.

Remark 3.8. Given a multigraph as above, consider the quiver (directed graph):

$$V+E \xleftarrow{h+\mathrm{id}} A+E \xrightarrow{g+t} E+V$$

where $h(i,e) = s(e)[i] \in V$ and $g(i,e) = e \in E$. A d-dimensional positional encoding representation is a representation of this quiver with constant dimension vector d.

4. T-MULTICATEGORIES

Multicategories are a special case of a more general construction, called T-multicategories, for a cartesian monad T, which we describe in this section. The case of multicategories corresponds to the monad $T = \mathsf{List}$.

4.1. **The bicategory of spans.** Recall from Section 1 that there is a category of spans where the objects are sets and the morphisms from V to W are given by spans $E \to V \times W$, which are the same as objects of the slice category over $V \times W$. We denote the collection of spans from V to W by $\mathsf{Span}(V,W)$. Composition is defined as

$$\begin{aligned} \mathsf{Span}(V,W) \times \mathsf{Span}(W,Z) &\to \mathsf{Span}(V,Z) \\ (E,F) &\mapsto E \times_W F \end{aligned}$$

where the maps $E \to V$ and $F \to Z$ make the fiber product $E \times_W F$ an object of the slice category over $V \times Z$. The associativity and unit axioms follow easily, where the units are the diagonal maps $V \to V \times V$. A key observation is that the composition defined above extends to a functor. Specifically, if $E \to E'$ and $F \to F'$ are maps of spans over $V \times W$ and $W \times Z$, respectively, then we have a map $E \times_W F \to E' \times_W F'$ of spans over $X \times Z$. The bicategory axioms[4] follow easily and we obtain the *bicategory of spans*.

Generalizing further, let T be an endofunctor on the category of sets. A T-span is an object of the slice category $\mathsf{Set}/TV \times W$; in other words, a set E with maps:

$$\begin{tikzcd}
& E \arrow[dl] \arrow[dr] & \\
TV & & W
\end{tikzcd}$$

---

[4] See Section 1.6 and Leinster [Lei04] for more on bicategories.

We denote the collection of T-spans from V to W by $\mathsf{Span}_T(V,W)$. If we naively try to compose spans $E \to TV \times W$ and $F \to TW \times Z$, we would form the fiber product $TE \times_{TW} F$. However, this maps to $T^2V \times Z$, so is actually a T^2-span rather than a T-span. Additionally, the unit for V would have to be a map $V \to TV \times V$; there is no obvious choice of a map $V \to TV$.

These considerations show that in order to compose spans, we need extra data associated to the endofunctor T. Namely, suppose that T is monad with unit and multiplication natural transformations:

$$\epsilon : \mathrm{id}_{\mathcal{C}} \to T \qquad \mu : T \circ T \to T$$

These need to satisfy the unit and associativity axioms:

$$\mu_X \circ \mu_{TX} = \mu_X \circ T\mu_X \qquad \mu_X \circ T\epsilon_X = \mathrm{id}_{TX} = \mu_X \circ \epsilon_{TX}$$

Then we can define a composition law:

$$\mathsf{Span}_T(V,W) \times \mathsf{Span}_T(W,Z) \to \mathsf{Span}_T(V,Z)$$

taking $E \to TV \times W$ and $F \to TW \times Z$ to the fiber product $TE \times_{TW} F$, defined by the following diagram:

(4.1)

$$\begin{tikzcd}
 & & & TE \times_{TW} F \arrow[dl] \arrow[dr] \arrow[dd, phantom, very near start, "\wedge"] & & \\
 & & TE \arrow[dl] \arrow[dr] & & F \arrow[dl] \arrow[dr] & \\
TV & T^2V \arrow[l, "\mu_V"'] & & TW & & Z
\end{tikzcd}$$

The proposed unit is $(\epsilon_V, \mathrm{id}) : V \to TV \times V$. Even though the fact that T is monad allows us to define the composition map and unit, it turns out that the unit and associativity coherence do not hold automatically; we introduce the notion of a cartesian monad in the next section in order to produce a bicategory.

4.2. **Cartesian monads.** We start with the main definition of this section.

Definition 4.1. Let $T = (T, \mu, \epsilon)$ be a monad on the category of sets. We say that T is a *cartesian monad* if:

- T is cartesian as a functor, so that T preserves pullbacks.
- μ and ϵ are cartesian as natural transformations, so that each of the naturality squares

$$\begin{tikzcd}
T^2V \arrow[r, "T^2f"] \arrow[d, "\mu_V"'] & T^2W \arrow[d, "\mu_W"] \\
TV \arrow[r, "Tf"] & TW
\end{tikzcd} \qquad \begin{tikzcd}
V \arrow[r, "f"] \arrow[d, "\epsilon_V"'] & W \arrow[d, "\epsilon_W"] \\
TV \arrow[r, "Tf"] & TW
\end{tikzcd}$$

  is a pullback for any morphism of sets $f : V \to W$.

Remark 4.2. A natural transformation between endofunctors of Set is cartesian if and only if the naturality square corresponding to the map $V \to 1$ to the terminal object is a pullback for any set V. The proof of this fact is an exercise.

Example 4.3. It is straightforward to verify that the following are cartesian monads:

(1) The monad List, also known as the free monoid monad.
(2) The monad List* of nonempty lists, also known as the free semigroup monad.
(3) For a set S, the monad $X \mapsto X + S$.
(4) A special case of the previous example is the maybe monad $X \mapsto X + 1$.
(5) For a monoid M, the monad $X \mapsto X \times M$.

Example 4.4. As a slightly more sophisticated example, let M be a commutative monoid and consider the endofunctor:

$$X \mapsto \mathsf{List}(X) \times M.$$

We claim that this is a cartesian monad. The unit is the map $x \mapsto ([x], 1)$, using the unit 1 of the monoid. The monadic multiplication is given by:

$$\mu_X : \mathsf{List}(\mathsf{List}(X) \times M) \times M \xrightarrow{\sim} \mathsf{List}(\mathsf{List}(X)) \times_{\mathbb{N}} \mathsf{List}(M) \times M \longrightarrow \mathsf{List}(X) \times M$$

where the last map concatenates $\mathsf{List}(\mathsf{List}(X)) \to \mathsf{List}(X)$ and multiplies $\mathsf{List}(M) \times M \to M$ using the monoid operation. The associativity of multiplication depends on M being commutative. The cartesian properties are now easy to verify using these definitions.

Example 4.5 (Non-cartesian nature of the Bag-monad)**.** The Bag monad fails to be cartesian in two ways. First, as an endofunctor, Bag is not cartesian; specifically, for sets X and Y, the natural map:

$$\mathsf{Bag}(\pi_1) \times \mathsf{Bag}(\pi_2) : \mathsf{Bag}(X \times Y) \to \mathsf{Bag}(X) \times_{\mathbb{N}} \mathsf{Bag}(Y)$$
$$\sum_{x,y} n_{x,y}(x,y) \mapsto \left(\sum_{x \in X} \sum_{y \in Y} n_{x,y} x \, , \, \sum_{y \in Y} \sum_{x \in X} n_{x,y} y \right)$$

is not a bijeciton in general. As a concrete illustration, let $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2\}$, and we exhibit two bags that map to the same pair of bags:

$$(x_1, y_1) + (x_2, y_2) \mapsto (x_1 + x_2, y_1 + y_2) \qquad (x_1, y_2) + (x_2, y_1) \mapsto (x_1 + x_2, y_1 + y_2)$$

One can also think about this geometrically. The set $\mathsf{Bag}(X)$ can be identified as the set of isomorphism classes of the category of bundles $E \to X$ over X with finite fibers. The non-cartesian nature of Bag reflects the fact that not every bundle on $X \times Y$ is the product of a bundle on X and a bundle on Y.

The second way in which the Bag monad fails to be carteisan is that μ is not cartesian. Indeed, taking $X = \{x, y\}$ and $Z = \{z\}$ consider the naturality square:

$$\begin{tikzcd}
\mathsf{Bag}(\mathsf{Bag}(X)) \arrow[r, "\mathsf{Bag}(\mathsf{Bag}(f))"] \arrow[d, "\mu_X"'] & \mathsf{Bag}(\mathsf{Bag}(Z)) \arrow[d, "\mu_1"] \\
\mathsf{Bag}(X) \arrow[r, "\mathsf{Bag}(f)"] & \mathsf{Bag}(Z)
\end{tikzcd}$$

where $f : X \to Z$ is the unique map. Define the bags of bags $\beta_1, \beta_2 \in \mathsf{Bag}(\mathsf{Bag}(X))$ and $\gamma \in \mathsf{Bag}(\mathsf{Bag}(Z))$ as characteristic functions of certain sets of bags:

$$\beta_1 = \chi_{\{x+y,x\}} \qquad \beta_2 = \chi_{\{2x,y\}} \qquad \gamma = \chi_{\{2z,z\}}$$

Then β_1 and β_2 both map to γ under $\mathsf{Bag}(\mathsf{Bag}(f))$, and to the bag $2x + y$ under μ_X, so the square above is not cartesian. (We note that the unit natural transformation ϵ is in fact cartesian.)

Example 4.6. The power set monad is also not cartesian. Neither the power set endofunctor, nor the unit natural transformation, nor the multiplication natural transformation is cartesian.

4.3. T-multicategories.

We now arrive at the main construction of this section.

Proposition 4.7. *Let T be a cartesian monad on the category* Set*. There is a bicategory* $\mathsf{Set}_{(T)}$ *where the objects are those of* Set*, and the category of morphisms from V to W is the slice category* $\mathsf{Span}_T(V, W)$*; composition is given by*

$$\begin{aligned} \mathsf{Span}_T(V,W) \times \mathsf{Span}_T(W,z) &\to \mathsf{Span}_T(V,Z) \\ E, F &\mapsto TE \times_{TW} F \end{aligned}$$

as in Diagram 4.1 above, and units given by $(\epsilon_V, \mathrm{id}) : V \to TV \times V$.

Sketch of proof. For associativity, let $E_1 \to TV \times W$, $E_2 \to TW \times Z$ and $E_3 \to TZ \times U$. Then:

$$\begin{aligned} T\left(TE_1 \times_{TW} E_2\right) \times_{TZ} E_3 &\xrightarrow{\sim} T^2E_1 \times_{T^2W} TE_2 \times_{TZ} E_3 \\ &\xrightarrow{\sim} TE_1 \times_{TW} T^2W \times_{T^2W} TE_2 \times_{TZ} E_3 \\ &\xrightarrow{\sim} TE_1 \times_{TW} TE_2 \times_{TZ} E_3 \end{aligned}$$

where the first map is $(T\pi_1, T\pi_2) \times \mathrm{id}$ and is an isomorphism due to the cartesian nature of T as functor, the second is $(\mu_E, T^2(E_1 \to W)) \times \mathrm{id} \times \mathrm{id}$ and is an isomorphism due to the cartesian nature of μ (so that $T^2E_1 = TE_1 \times_{TW} T^2W$), and the last due to eliminating T^2W. The left unit axiom is straightforward to verify, and doesn't require T to be carteisan. For the right unit axiom, take $(s,t) : E \to TV \times W$ and consider the commutative diagram:

$$\begin{tikzcd}
 & & E \arrow[rr, "(\epsilon_E{,}t)"] \arrow[dl, "s"'] \arrow[dr, "\epsilon_E"] & & TE_{TW}W \arrow[dl] \arrow[dr] \arrow[d, phantom, "\wedge" near start] & & \\
 & TV \arrow[dl, "\mathrm{id}"'] \arrow[dr, "\epsilon_{TV}"] & & TE \arrow[dl, "Ts"'] \arrow[dr, "Tt"] & & W \arrow[dl, "\epsilon_W"'] \arrow[dr, "\mathrm{id}"] & \\
TV & & T^2V \arrow[ll, "\mu_V"'] & & TW & & W
\end{tikzcd}$$

We examine this diagram in detail. First, the map $(\epsilon_E, Tt) : E \to TE \times_{TW} W$ is an isomorphism by the hypothesis that ϵ is cartesian; we show that it is an isomorphism in the slice category over $TV \times W$. This follows from the commutativity of the lower left triangle (unit axiom for T), and the commutativity of the left square (ϵ is a natural transformation). $\square$

Definition 4.8. Let T be a caretesian monad on Set. A *T-multicategory* is a monad in the bicategory $\mathsf{Set}_{(T)}$; that is, it consists of a morphism $C \to TV \times V$ together with maps:

$$\eta : V \to C, \qquad m : TC \times_{TV} C \to C$$

in the slice category such that the following diagrams commute:

$$\begin{tikzcd}
T(TC \times_{TV} C) \times_{TV} C \arrow[r, "\sim"] \arrow[d, "Tm\times\mathrm{id}"'] & TC \times_{TV} TC \times_{TV} C \arrow[r, "\mathrm{id}\times m"] & TC \times_{TV} C \arrow[d, "m"] \\
TC \times_{TV} C \arrow[rr, "m"] & & C
\end{tikzcd}$$

$$\begin{tikzcd}
C \arrow[r, "\sim"] \arrow[drr, "\mathrm{id}"'] & TC \times_{TV} V \arrow[r, "\mathrm{id}\times\eta"] & TC \times_{TV} C \arrow[d, "m"'] & TV \times_{TV} C \arrow[l, "T\eta\times\mathrm{id}"'] & C \arrow[l, "\sim"'] \arrow[dll, "\mathrm{id}"] \\
& & C & &
\end{tikzcd}$$

In the top diagram, the first map of the top composition is $(\mu_C \circ T\pi_1, T\pi_2) \times \mathrm{id}$ which is the associativity isomorphism of the composition of 1-cells, and uses the cartesian nature of the monad T. Similarly, the isomorphism $C \to TC \times_{TV} V$ uses the cartesian nature of the natural transformation ϵ.

In the case $V = 1$ is a point, we obtain a *T-operad*.

Example 4.9. We examine examples:

(1) $T = \mathrm{id}$. A T-multicategory is a small category. A T-operad is a monoid.
(2) $T = \mathsf{List}$. A T-multicategory is a (usual) multicategory. A T-operad is a (usual) operad.
(3) $T = \mathsf{List}^*$. A T-multicategory is a (usual) multicategory with no nullaries.
(4) $T = - + S$. A T-multicategory is a small category together with functors $Y_s : \mathcal{C} \to \mathsf{Set}$ for $s \in S$.
(5) $T = - \times M$ for a monoid M. A T-multicategory is a small category together with a functor $\mathcal{C} \to \bullet/M$.
(6) $T = \mathsf{List}(-) \times M$ for a commutative monoid M. A T-multicategory is a small multicategory together with a functor to the multicategory with one object defined by M (see Example 3.3; every multimorphism set is M, regardless of the length of the input, and composition uses the monoid operation).

Example 4.10 (Free T-multicategory)**.** Let $E \to TV \times V$ be an endomorphism 1-cell in the bicategory $\mathsf{Set}_{(T)}$. The free T-multicategory generated by E has 1-cell given by the initial algebra A of the endofunctor $X \mapsto V + TX \times_{TV} E$ of the category $\mathsf{Set}/TV \times V$. (The initial algebra exists as long as T is finitary.) The unit 2-cell is the inclusion

$$\eta : V \xrightarrow{\text{left}} V + TA \times_{TV} E \xrightarrow{\sim} A$$

To define the multiplication 2-cell, first observe that the initial algebra admits a filtration:

$$A^{(0)} = V, \qquad A^{(n)} = V + T\left(A^{(n-1)}\right) \times_{TV} E \qquad \text{for } n \geq 1.$$

where the inclusions $A^{(n-1)} \hookrightarrow A^{(n)}$ are defined by induction starting with $A^{(0)} = V \hookrightarrow V + E = A^{(1)}$. Now define the composition 2-cell by induction on the filtered pieces as follows. First, the map $(\epsilon_A, t_A) : A \to TA \times_{TV} V = TA \times_{TV} A^{(0)}$ is an isomorphism due to ϵ being cartesian; set $m^{(0)}$ to be the inverse of this map. For $n \geq 1$, the composition map $m^{(n)}$ is defined via the composition:

$$\begin{aligned} TA \times_{TV} A^{(n)} &\xrightarrow{\sim} TA \times_{TV} \left(V + T(A^{(n-1)}) \times_{TV} E\right) \\ &\xrightarrow{\sim} TA \times_{TV} V + TA \times_{TV} T(A^{(n-1)}) \times_{TV} E \\ &\xrightarrow{\sim} A + T(TA \times_{TV} A^{(n-1)}) \times_{TV} E \\ &\longrightarrow A + TA \times_{TV} E \xrightarrow{\sim} A + A \longrightarrow A \end{aligned}$$

where the first map uses the definition of $A^{(n)}$; the second map distributes products over sums; the third map invokes the cartesian property of the monad T; the fourth map is $1 + Tm^{(n-1)} \times 1$, which uses the induction hypothesis; the fifth map references the initiality of A; and the last map is the obvious fold map. It remains to verify the unit and associativity axioms for (A, η, m) as a monad in the bicategory $\mathsf{Set}_{(T)}$. The unit axioms are straightforward; the associativity axiom is technical and we give a full proof in Appendix B.

Example 4.11. We examine specific free T-categories.

1. $T = \mathrm{id}$. An endomorphism 1-cell in $\mathsf{Set}_{(T)}$ is nothing more than a directed graph: $E \to V \times V$. The corresponding free T-multicategory is free category generated by this graph as discussed above. As a special case, if the directed graph is $(1, f) : V \to V \times V$ for an endomorphism $f : V \to V$, then the set of morphisms of the free category is $V \times \mathbb{N}$ with map to $V \times V$ given by $(v, f^n(v))$.
2. $T = \mathsf{List}$. An endomorphism 1-cell in $\mathsf{Set}_{(T)}$ is a multigraph $E \to \mathsf{List}(V) \times V$, and the corresponding free T-multicategory is the usual free multicategory generated by the directed graph. Similar comments hold for List^*.
3. $T = - + S$. An endomorphism 1-cell in $\mathsf{Set}_{(T)}$ is a map $(d, t) : E \to (V + S) \times S$. Set $E_1 = d^{-1}(V)$ and $Y_s(v) = d^{-1}(s) \cap t^{-1}(v) \subseteq E$ for $s \in S$ and $v \in V$, so that $E = E_1 + \sum_s \sum_v Y_s(v)$. The corresponding free T-multicategory is the free category generated by the directed graph defined by the edges E_1, with and S-family of functors to Set given on objects by the $Y_s(v)$, and on morphsims in a straightforward way.
4. $T = - \times M$ for a monoid M. An endomorphism 1-cell in $\mathsf{Set}_{(T)}$ is a directed graph with edge labels in M, that is, $E \to V \times M \times V$. The corresponding free T-multicategory is the free category generated by the direct graph with functor to the one-object category $\bullet/M$ defined by using the multiplication in M to extend labels from edges to paths. Similar comments hold for the monad $\mathsf{List}(-) \times M$, with M commutative.

4.4. **Composition of monads.** As a final section, we record observations about the composition of monads. Let $(T_1, \mu^{(1)}, \epsilon^{(1)})$ and $(T_2, \mu^{(2)}, \epsilon^{(2)})$ be monads on the category of sets. Suppose

$$\sigma : T_1 \circ T_2 \to T_2 \circ T_1$$

is a natural transformation such that the following diagrams commute for any set X:

$$\begin{tikzcd}
 & T_1T_2X \arrow[dd, "\sigma_X"] & \\
T_1X \arrow[ur, "T_1\mu^{(2)}_X"] \arrow[dr, "\mu^{(2)}_{T_1X}"'] & & T_2X \arrow[ul, "\mu^{(1)}_{T_2X}"'] \arrow[dl, "T_2\mu^{(1)}_X"] \\
 & T_2T_1X &
\end{tikzcd}$$

$$\begin{tikzcd}
T_1^2T_2X \arrow[r, "T_1\sigma_X"] \arrow[d, "\mu^{(1)}_{T_2X}"'] & T_1T_2T_1X \arrow[r, "\sigma_{T_1X}"] & T_2T_1^2X \arrow[d, "T_2\mu^{(1)}_X"] \\
T_1T_2X \arrow[rr, "\sigma_X"] & & T_2T_1X
\end{tikzcd}$$

$$\begin{tikzcd}
T_1T_2^2X \arrow[r, "\sigma_{T_2X}"] \arrow[d, "T_1\mu^{(2)}_X"'] & T_2T_1T_2X \arrow[r, "T_2\sigma_X"] & T_2^2T_1X \arrow[d, "\mu^{(2)}_{T_1X}"] \\
T_1T_2X \arrow[rr, "\sigma_X"] & & T_2T_1X
\end{tikzcd}$$

Then $T_2 \circ T_1$ is monad with unit defined by either of the compositions in the following commutative diagram:

$$\begin{tikzcd}
X \arrow[r, "\epsilon^{(1)}_X"] \arrow[d, "\epsilon^{(2)}_X"'] & T_1X \arrow[d, "\epsilon^{(2)}_{T_1X}"] \\
T_2X \arrow[r, "T_2\epsilon^{(1)}_X"] & T_2T_1X
\end{tikzcd}$$

and multiplication defined by either of the compositions in the following diagram:

$$\begin{tikzcd}
T_2T_1T_2T_1X \arrow[r, "T_2\sigma_{T_1X}"] & T_2^2T_1^2X \arrow[r, "T_2^2\mu^{(1)}_X"] \arrow[d, "\mu^{(2)}_{T_1^2X}"'] & T_2^2T_1X \arrow[d, "\mu^{(2)}_{T_1X}"] \\
 & T_2T_1^2X \arrow[r, "T_2\mu^{(1)}_X"] & T_2T_1X
\end{tikzcd}$$

Moreover, if T_1 and T_2 are cartesian monads, and σ is a cartesian natural transformation, then T_2T_1 is a cartesian monad as well.

Example 4.12. Here are some examples of cartesian monads and their composition:

(1) We have already discussed above the example of $T_1 = \mathsf{List}$ and $T_2 = - \times M$, for a commutative monoid M. In this case, the natural transformation $\mathsf{List}(X \times M) \to \mathsf{List}(X) \times M$ is known as a *sequence map*, which is terminology used in Haskell.

(2) There is also a sequence map $\mathsf{List}(X+S) \to \mathsf{List}(X) + S$ that takes $[k_1, \dots, k_n]$ to $[x_1, \dots, x_n]$ if $k_i = x_i \in X$ for all i and otherwise to $s = k_i$ where i is minimal such that $s \in k_i$. However, this natural transformation is not cartesian.

(3) Other non-examples are:

$$\mathsf{List}(X) \times M \to \mathsf{List}(X \times M), \qquad ([x_1, \dots, x_n], m) \mapsto [(x_1, m), \dots, (x_n, m^n)]$$

$$\mathsf{List}(X) + S \to \mathsf{List}(X+S), \qquad k \mapsto \begin{cases} \mathbf{x} & \text{if } k = \mathbf{x} \in \mathsf{List}(X) \\ [x] & \text{if } k = s \in S \end{cases}$$

(4) We can take $T_1 = - \times M$ and $T_2 = - + S$ with:

$$\sigma_X : (X+S) \times M = S \times M + S \times M \longrightarrow X \times M + S$$

given by the identity in the $X+S$ cofactor and the projection map on the $S \times M$ factor. The natural transformation σ is cartesian, and satisfies the commutativity axioms.

(5) We can take $T_1 = - + S$ and $T_2 = - \times M$ with:

$$\sigma_X : X \times M + S \longrightarrow X \times M + S \times M = (X+S) \times M$$

given by the identity in the $X+S$ cofactor and the inclusion $s \mapsto (s, 1)$ in the S factor, where 1 is the unit of the monoid M. The natural transformation σ is cartesian, and satisfies the commutativity axioms.

5. Bag-graphs

In this section, we discuss the relationship between multigraphs and Bag-graphs. First, we collect some facts about non-cartesian monads.

5.1. **Cartesian and non-cartesian monads.** Suppose T is a monad on Set.

Definition 5.1. A T-graph on the vertex set V is an object $E \to TV \times V$ of the slice category over $TV \times V$.

Any set V is a T-graph via $(\epsilon_V, \mathrm{id}) : V \to V \times V$, where ϵ_V is the monadic unit of T applied to V. Moreover, given a T-graph $E \to TV \times V$, we have an endofunctor of $\mathsf{Set}/TV \times V$ given by:

$$X \mapsto V + TX \times_{TV} E$$

The T-graph structure on $TX \times_{TV} E$ uses the monadic multiplication $T^2 V \xrightarrow{\mu_V} TV$. As long as T is finitary, this endofunctor has an initial algebra determined by the filtration:

$$A^{(0)} = V, \qquad A^{(n)} = V + TA^{(n-1)} \times_{TV} E$$

for $n \geq 1$. There are glueing and deglueing isomorphisms:

$$\mathsf{glue} : V + TA \times_{TV} E \xrightarrow{\sim} A \qquad\qquad \mathsf{deglue} : A \xrightarrow{\sim} V + TA \times_{TV} E$$

However, if T fails to be cartesian, then there may not be a well-defined composition map $TA \times_{TV} A \to A$. In the following sections, we examine the case the non-cartesian monad $T = \mathsf{Bag}$ in more detail.

5.2. **Quasi-multimorphisms.** In a multigraph, each edges has a list of inputs; the fact that a list is ordered is crucial in defining the composition in the free multicategory. What happens if we relax the assumption that the inputs of an edge is an ordered list? Specifically, we can replace each input list with a bag of inputs.

Definition 5.2. A Bag-graph consists of a vertex set V, and edge set E, and a map $\Delta : E \to \mathsf{Bag}(V) \times V$. The set of quasi-multimorphisms $\mathsf{QMMor} = \mathsf{QMMor}_\Delta$ for the Bag-graph Δ is defined as the initial algebra of the endofunctor:

$$F : X \mapsto V + \mathsf{Bag}(X) \times_{\mathsf{Bag}(V)} E$$

of the slice category over $\mathsf{Bag}(V) \times V$. The set QMMor is naturally filtered via:

$$\begin{aligned} \mathsf{QMMor}_{\leq 0} &= V \\ \mathsf{QMMor}_{\leq d} &= V + \mathsf{Bag}(\mathsf{QMMor}_{\leq d-1}) \times_{\mathsf{Bag}(V)} E \qquad\qquad \text{for } d \geq 1 \end{aligned}$$

so that $\mathsf{QMMor}_{\leq d} = F^{d+1}(\varnothing)$.

Quasi-multimorphisms also admit an interpretation as trees where the d-th filtered piece is the set of trees of depth at most d; specifically:

- Depth zero trees are elements of V via $\mathsf{QMMor}_{\leq 0} = F\varnothing = V$.
- Depth one trees are elements of E via $\mathsf{QMMor}_{\leq 1} \setminus \mathsf{QMMor}_{\leq 0} = E$.
- Given an edge e with source list $s \in \mathsf{Bag}(V)$, and given, for each $v \in V$, $s(v)$ elements of QMMor of depth at most $d-1$, we can attach the quasi-multimorphisms to e to produce a tree of depth at most d.

5.3. **From multimorphisms to quasimultimorphisms.** The forgetful natural transformation $\mathsf{List} \to \mathsf{Bag}$ takes a list to the bag counting multiplicities of elements in the list, and allows us to "forget" any multigraph to a Bag-graph.

With this in mind, let $\Gamma = (V, E)$ be a multigraph and abbreviate MMor_Γ by just MMor. Regarding Γ as a Bag-graph, abbreviate QMMor_Γ by just QMMor quasi-multimorphisms. We define a map

$$q : \mathsf{MMor} \to \mathsf{QMMor}$$

by induction on the filtered pieces as follows. The base step is the identity map:

$$\mathsf{MMor}_{\leq 0} = V \xrightarrow{\text{id}} V = \mathsf{QMMor}_{\leq 0}$$

For the induction step, given $\mathsf{MMor}_{\leq d-1} \to \mathsf{QMMor}_{\leq d-1}$, we extend to the next filtered piece using the composition:

$$\begin{aligned} \mathsf{MMor}_{\leq d} = V + \mathsf{List}(\mathsf{MMor}_{\leq d-1}) \times_{\mathsf{List}(V)} E &\to V + \mathsf{List}(\mathsf{QMMor}_{\leq d-1}) \times_{\mathsf{List}(V)} E \\ &\to V + \mathsf{Bag}(\mathsf{QMMor}_{\leq d-1}) \times_{\mathsf{Bag}(V)} E = \mathsf{QMMor}_{\leq d} \end{aligned}$$

where the first map uses the induction hypotheses, and the second uses the forgetful natural transformation from List to Bag.

5.4. **Equivalence relation.** Continuing the set-up from the previous section, we define an equivalence relation on the set MMor of multimorphisms by induction as follows.

- The equivalence class of the identity id_v, for $v \in V$, consists of only itself.
- The equivalence class of the composition
$$\mathsf{compose}\left([f_1, \ldots, f_n], e\right)$$
consists of all compositions of the form:
$$\mathsf{compose}\left(\left[f_{\sigma(1)}, \ldots, f_{\sigma(n)}\right], e\right)$$
where $\sigma \in S_n$ is a permutation such that $f_{\sigma(i)}$ belongs to the equivalence class of f_i for each $i = 1, \ldots, n$.

In other words, we can permute inputs to a composition that are equivalent, with the base case being the identity multimorphisms, which are posited to be pairwise non-equivalent. A consequence is that the edges, as the set of depth-one multimorphisms, are also pairwise non-equivalent.

Proposition 5.3. *The target map* $\mathsf{MMor} \to V$ *is invariant under the equivalence relation, as is the source map composed with the forgetful map to* $\mathsf{Bag}(V)$*:*
$$\mathsf{MMor} \to \mathsf{List}(V) \to \mathsf{Bag}(V)$$
Moreover, the map $\mathsf{MMor} \to \mathsf{QMMor}$ *is precisely the quotient map by the equivalence relation* $\sim$*.*

Sketch of proof. The first two claims are straightforward to verify, and together imply that the set of equivalence classes naturally belongs to the slice category over $\mathsf{Bag}(V) \times V$. The last claim can be shown by induction on the filtered pieces. □

We comment on positional encodings of quasimultimorphisms. Recall that a d-dimensional positional encoding representation of a multigraph $\Gamma : E \to V \times V$ is a map $\rho : A + E \to \mathbb{R}^{d \times d}$, where $A = (\sum_{n \in \mathbb{N}} [n]) \times_{\mathbb{N}} E$ is the set of arguments. Such a representation gives rise to a d-dimensional encoded traversal:
$$\Psi : \mathsf{MMor}_\Gamma \times \mathbb{R}^d \to \mathsf{List}\left(\mathbb{R}^d\right)$$

Definition 5.4. A d-dimensional positional encoding representation is said to be *symmetric* if $\rho(i, e) = \rho(j, e)$ for any $(i, e), (j, e) \in A$ with $s(e)[i] = s(e)[j]$.

The following lemma is easy to verify:

Lemma 5.5. *The encoded traversal of a symmetirc positional encoding representation factors through* QMMor_Γ*:*
$$\begin{tikzcd}
\mathsf{MMor}_\Gamma \times \mathbb{R}^d \arrow[r, "\Psi"] \arrow[d, "q"'] & \mathsf{List}\left(\mathbb{R}^d\right) \\
\mathsf{QMMor}_\Gamma \times \mathbb{R}^d \arrow[ur] &
\end{tikzcd}$$

5.5. **Failure of composition.** We claim that there is no well-defined composition map of quasimultimorphisms. In other words, the glueing map $V + \mathsf{Bag}(\mathsf{QMMor}) \times_{\mathsf{Bag}(V)} E \to \mathsf{QMMor}$ does not extend to one with E replaced by QMMor.

Lemma 5.6. *There is no map*

$$\mathsf{Bag}(\mathsf{QMMor}) \times_{\mathsf{Bag}(V)} \mathsf{QMMor} \longrightarrow \mathsf{QMMor}$$

making the following diagram commute:

$$\begin{tikzcd}
\mathsf{List}(\mathsf{MMor}) \times_{\mathsf{List}(V)} \mathsf{MMor} \arrow[r, "\mathsf{compose}"] \arrow[d] & \mathsf{MMor} \arrow[d] \\
\mathsf{Bag}(\mathsf{QMMor}) \times_{\mathsf{Bag}(V)} \mathsf{QMMor} \arrow[r, "\times" description] & \mathsf{QMMor}
\end{tikzcd}$$

Proof. We prove the lemma by providing a counterexample. Let $\Gamma = (V = \{v\}, E = \{L, N\})$ be the binary tree multigraph, where the edge L (for "leaf") is nullary, and the edge N (for "node") is binary. Consider the following multimorphism:

$$f := \mathsf{compose}\left([N, v], N\right)$$

Observe that the images of $([L, v, v], f)$ and $([v, v, L], f)$ under the left vertical map are the same. However, their images under the composition map are, respectively:

$$\begin{aligned}
\mathsf{compose}\left([L, v, v], f\right) &= \mathsf{compose}([\mathsf{compose}([L, v], N), v], N) \\
\mathsf{compose}\left([v, v, L], f\right) &= \mathsf{compose}([N, L], N)
\end{aligned}$$

which are not equivalent, and hence have different images in QMMor. □

6. POLYNOMIALS

In this section, we describe another combinatorial description of multigraphs and Bag-graphs in terms of polynomial spans. This perspective relates to polynomial functors, which we define precisely.

6.1. **Polynomial spans.** We begin directly with the definition of a polynomial span:

Definition 6.1. A *finitary polynomial span* from V to W consists of sets A and E and maps:

$$V \leftarrow A \rightarrow E \rightarrow W$$

where $A \to E$ has finite fibers. A *finitary polynomial span with fiber orderings* is a finitary polynomial span

$$V \leftarrow A \rightarrow E \rightarrow W$$

together with bijections $[|A_e|] \to A_e$ for every $e \in E$. If $V = W$, we have a *finitary polynomial endospan* (resp. *with fiber orderings*) of V.

Remark 6.2. One reason for the use of "polynomial" in the above definition is that there is a correspondence between polynomials in k variables and finitary polynomial spans from $V = [k]$ to $W = [1]$. See Section 6.4 for more details.

Lemma 6.3. *There are one-to-one correspondences:*

$$\{\textit{finitary polynomial spans from } V \textit{ to } W\} \leftrightarrow \{\textit{objects of } \mathsf{Set}/\mathsf{Bag}(V) \times W\}$$

$$\{\textit{finitary polynomial spans with fiber orderings from } V \textit{ to } W\} \leftrightarrow \{\textit{objects of } \mathsf{Set}/\mathsf{List}(V) \times W\}$$

Sketch of proof. Given a finitary polynomial span $V \leftarrow A \rightarrow E \rightarrow W$, the set E defines an object of the slice category over $\mathsf{Bag}(V) \times W$; indeed, we already have a map to W, while the map to $\mathsf{Bag}(V)$ is given by:

$$e \mapsto [v \mapsto |A_e \cap A_v|]$$

where A_e denotes the fiber of $A \rightarrow E$ over e and similarly for A_v. Conversely, given a morphism $(s,t) : E \rightarrow \mathsf{Bag}(V) \times W$, we set $A = \sum_{v\in V} \sum_{e\in E} [s(e)(v)]$, which has cofactor projections to V and E. The fibers of the map $A \rightarrow E$ are finite since, for each $e \in E$, there are only finitely many v such that $s(e)(v) \neq 0$.

Now, given a finitary polynomial endospan $V \overset{h}{\leftarrow} A \rightarrow E \rightarrow V$ with fiber orderings, the set E defines an object of the slice category over $\mathsf{List}(V) \times W$, where the map $E \rightarrow \mathsf{List}(V)$ is given by:

$$e \mapsto [h \circ \phi(1), \dots, h \circ \phi(n)]$$

where $n = |A_e|$ and $\phi : [n] \rightarrow A_e$ is the ordering of the fiber. On the other hand, given an object $(s,t) : E \rightarrow \mathsf{List}(V) \times W$ of the slice category, the set of arguments $A = (\sum_{n\in\mathbb{N}} [n]) \times_{\mathbb{N}} E$ (see Section 3.5) allows us to obtain a polynomial span. The ordering on each $[n]$ gives an ordering on each fiber of $A \rightarrow E$. ☐

Remark 6.4. By defining a category of finitary polynomial spans, one can extend these correspondences to equivalences of categories.

Remark 6.5. In the case of multigraphs, an argument $(i,e) \in A$ indicates a index of the source list of an edge. In the case of Bag-graphs, an argument $a \in [s(e)(v)]$ indicates what number of v-typed sources of the edge e are filled, where the maximum is $s(e)(v)$.

In the case $W = V$, we have that finitary polynomial endospans of V correspond exactly to Bag-graphs on the vertex set V, while finitary polynomial endospans of V with fiber orderings correspond exactly to multigraphs on the vertex set V.

6.2. **Dependent sum and product functors.** In order to define polynomial functors, we now make a digression to define the dependent sum and product functors. Suppose $f : X \rightarrow Y$ is a morphism in the category $\mathcal{C} = \mathsf{Set}$ of sets (on can generalize to any locally closed cartesian category). We have the following functors on the slice categories:

- Pullback:

$$\begin{array}{rcl} f^* : \mathcal{C}/Y & \rightarrow & \mathcal{C}/X \\ T = \sum_{y\in Y} T_y & \mapsto & T \times_Y X = \sum_{x\in X} T_{f(x)} \end{array}$$

- Dependent sum:

$$\begin{array}{ccc} \Sigma_f : \mathcal{C}/X & \to & \mathcal{C}/Y \\ S = \sum_{x \in X} S_x & \mapsto & S = \sum_{y \in Y} \left(\sum_{x \in f^{-1}(y)} S_x \right) \end{array}$$

- Dependent product:

$$\begin{array}{ccc} \Pi_f : \mathcal{C}/X & \to & \mathcal{C}/Y \\ S = \sum_{x \in X} S_x & \mapsto & \sum_{y \in Y} \left(\prod_{x \in f^{-1}(y)} S_x \right) \end{array}$$

The dependent sum Σ_f and the dependent product Π_f are the left adjoint and right adjoints, respectively, to the pullback f^*. We give the argument for Σ_f:

$$\begin{aligned} \mathrm{Hom}_{\mathcal{C}/Y}\left(\Sigma_f S \ , \ T\right) &= \mathrm{Hom}_{\mathcal{C}/Y}\left(\sum_{y \in Y} \sum_{x \in f^{-1}(y)} S_x \ , \ \sum_{y \in Y} T_y \right) \\ &= \prod_{y \in Y} \mathrm{Hom}_{\mathcal{C}} \left(\sum_{x \in f^{-1}(y)} S_x \ , \ T_y \right) \\ &= \prod_{y \in Y} \prod_{x \in f^{-1}(y)} \mathrm{Hom}_{\mathcal{C}}\left(S_x \ , \ T_y\right) \\ &= \prod_{x \in X} \mathrm{Hom}_{\mathcal{C}}\left(S_x \ , \ T_{f(x)}\right) \\ &= \mathrm{Hom}_{\mathcal{C}/X}\left(\sum_{x \in X} S_x \ , \ \sum_{x \in X} T_{f(x)} \right) \\ &= \mathrm{Hom}_{\mathcal{C}/X}\left(S \ , \ f^* T\right) \end{aligned}$$

where the first and sixth equalities follows from definitions, the second and fifth from the nature of morphisms in the slice category, the third from the fact that colimits in the first factor turn into limits outside the Hom, with disjoint union in the first factor, and the fourth from reindexing. The proof of the adjunction (f^*, Π_f) is similar, the relevant step being:

$$\mathrm{Hom}_{\mathcal{C}}\left(T_y, \prod_{x \in f^{-1}(y)} S_x \right) = \prod_{x \in f^{-1}(y)} \mathrm{Hom}_{\mathcal{C}}\left(T_y, S_x\right)$$

where $T = \sum_{y \in Y} T_y$ and $S = \sum_{x \in X} S_x$ are objects of the respective slice category.

Remark 6.6. Given an object $S \to X$ of the slice category $\mathcal{C}/X$, taking its image in X defines a functor $\mathcal{C}/X \to \mathcal{P}(X)$ to the poset category of subsets of X with inclusions. For a map $f : X \to Y$, the pullback f^* corresponds to the inverse image functor $f^{-1}(B) = \{x \in X \mid f(x) \in B\}$, the dependent sum Σ_f to the direct image functor $f_*(A) = \{y \in$

$Y \mid \exists x \in f^{-1}(y) : x \in A\}$, and the dependent product Π_f to the shriek image functor $f_!(A) = \{y \in Y \mid \forall x \in f^{-1}(y) : x \in A\}$.

6.3. **Polynomial functors.** Let $V \xleftarrow{h} A \xrightarrow{g} E \xrightarrow{t} W$ be a polynomial span (not neceesarily finitary). The functor on the slice categories

$$\Sigma_t \circ \Pi_g \circ h^* : \mathsf{Set}/V \to \mathsf{Set}/W$$

is known as a *polynomial functor*. It has the explicit form:

$$\begin{aligned}\Sigma_t \circ \Pi_g \circ h^*(S) &= \sum_{w \in W} \sum_{e \in E_w} \prod_{a \in A_e} S_{h(a)} \\ &= \sum_{w \in W} \sum_{e \in E_w} \prod_{v \in V} S_v^{A_e \cap A_v}\end{aligned}$$

where $A_e = g^{-1}(e)$, $A_v = h^{-1}(v)$, and $E_w = t^{-1}(w)$.

Remark 6.7. One can take the definition of polynomial functors as those arising from polynomial spans. Alternatively, polynomial endofunctors on Set can be defined as the smallest class of functors containing the identity functor and constant functors, and closed under taking arbitrary coproducts and constant exponentials. See [Jac16, Section 2.2]

6.4. **Ordinary polynomials.** We begin with the elementary observation that, for any integer $k \geq 0$, there is a bijection:

$$\mathbb{N}^k \xrightarrow{\sim} \mathsf{Bag}([k]) \tag{6.1}$$

taking $\mathbf{n} = (n_1, \ldots, n_k)$ to the bag sending i to n_i. Moreover, there is a bijection

$$\begin{aligned}\mathbb{N}[x_1, \ldots, x_k] &\xrightarrow{\sim} \mathsf{Bag}(\mathbb{N}^k) \\ \sum_{\mathbf{n} \in \mathbb{N}^k} f_{\mathbf{n}} x_1^{n_1} \cdots x_k^{n_k} &\longmapsto [\mathbf{n} \mapsto f_{\mathbf{n}}]\end{aligned}$$

which extracts the coefficients from a polynomial in k variables with coefficients in $\mathbb{N}$, noting that finitely many of the coefficients are nonzero. Combining with the bijectionn of Equation 6.1, we have a commutative diagram:

$$\begin{tikzcd}\mathbb{N}^k \arrow[r, "\sim"] \arrow[d] & \mathsf{Bag}([k]) \arrow[d] \\ \mathbb{N}[x_1, \ldots, x_k] \arrow[r, "\sim"] & \mathsf{Bag}(\mathsf{Bag}([k]))\end{tikzcd}$$

where the left vertical map sends $\mathbf{n} = (n_1, \ldots, n_k)$ to the monomial $x_1^{n_1} \cdots x_k^{n_k}$, and the right vertical map is the unit of the Bag monad applied to $\mathsf{Bag}([k])$.

Remark 6.8. The monadic multiplication map $\mathsf{Bag}(\mathsf{Bag}(k)) \to \mathsf{Bag}([k])$ can be identified with the map sending a polynomial f to the element of $\mathbb{N}^k$ whose i-component is the partial derivative of f with respect to x_i, evaluated at $(1, \ldots, 1)$.

Now let V be a finite set (if desired, one can identify V with $[k]$). To every map $f : W \to \mathsf{Bag}(\mathsf{Bag}(V))$ there is a corresponding polynomial span

$$V \leftarrow A \to E \to W$$

with $E = \sum_{w \in W} \sum_{b \in \mathsf{Bag}([k])} [f(w)(b)]$ and $A = \sum_{w \in W} \sum_{b \in \mathsf{Bag}([k])} \sum_{v \in V} [f(w)(b)] \times [b(v)]$ where the maps are the cofactor projections maps. Thus, at least when V is finite, every W-family of polynomials in V gives rise to a polynomial span. However, there are polynomial spans that do not correspond so neatly to usual polynomials. We invite the reader to determine which of the multigraphs in Section 3.4 correspond to polynomials, and to compute those polynomials. For example, the multigraph whose terms are binary trees corresponds to the polynomial $1 + x^2$.

REFERENCES

[AMMU15] Jiří Adámek, Stefan Milius, Lawrence S. Moss, and Henning Urbat, *On finitary functors and their presentations*, Journal of Computer and System Sciences **81** (August 2015), no. 5, 813–833.

[Jac16] Bart Jacobs, *Introduction to Coalgebra: Towards Mathematics of States and Observation*, Cambridge University Press, 2016.

[Lei04] Tom Leinster, *Higher Operads, Higher Categories*, London Mathematical Society Lecture Note Series, Cambridge University Press, Cambridge, 2004.

Appendix A. The map q

A.1. **Group action.** Let $\Gamma = (V, E)$ be a multigraph. We abbreviate MMor_Γ by just MMor. One may be tempted to define the equivalence relation on MMor defined in Section 5.4 in terms of a group action as follows (we will see that this attempt is too simple). For each edge, set:

$$G_e = \{\sigma \in S_n \mid s(e)_i = s(e)_{\sigma(i)}\}$$

where n is the arity of e. In other words, G_e is the stabilizer of the source list $s(e) \in V^n$ of e under the action of the symmetric group S_n on V^n by permutation of factors. Then define an action of the group $G = \prod_{e \in E} G_e$ on MMor as follows:

- The action of G on each identity multimorphism is trivial.
- The action of $g = (\sigma_e)_{e \in E} \in G$ takes the composition:

  $$\mathsf{compose}\left([f_1, \ldots, f_n], e'\right)$$

  to the composition:

  $$\mathsf{compose}\left(\left[f_{\sigma_{e'}(1)}, \ldots, f_{\sigma_{e'}(n)}\right], e'\right)$$

In other words, the only factor of $G = \prod_{e \in E} G_e$ that acts non-trivially on the composition is the one corresponding to the head edge. It is the case that any two multimorphisms in the same orbit are in the same equivalence class as defined above. The converse, however, is not true since the group action does not affect the deeper layers of the tree. One would need a larger group and a more refined action that we do not discuss here.

A.2. **Endofunctors.** We begin with some general results about endofunctors and initial algebras. Let F and $\overline{F}$ be endofunctors of categories C and D, respectively.

Proposition A.1. *Suppose $J : C \to D$ is a full and essentially surjective functor, and that $\eta : J \circ F \to \overline{F} \circ J$ is a natural isomorphism. If a is an initial algebra for F, then Ja is an initial algebra for $\overline{F}$.*

Sketch. To sketch the proof of this claim: one shows that (1) Ja is an algebra for $\overline{F}$, and (2) given any $\overline{F}$-algebra x, there is an $\overline{F}$-algebra homomorphism from Ja to x. The latter step uses the essential surjectivity of J to produce an object $\tilde{x}$ of C such that $J\tilde{x}$ is isomorphic to x. Using η and the fact that J is full, one argues that $\tilde{x}$ is an F-algebra, and hence admits an F-algebra homomorphism from a. Applying J to this homomorphism and using η, one arrives at an $\overline{F}$-algebra homomorphism $Ja \to x$. □

The proof of the following result is straightforward:

Proposition A.2. *Suppose $\nu : G_1 \to G_2$ is a natural transformation of endofunctors of a category C. There is a functor from G_2-algebras to G_1-algebras taking $G_2 b \to b$ to $G_1 b \xrightarrow{\nu_b} G_2 b \to b$.*

In particular, if a and b are initial F- and G-algebras, respectively, then there is an induced F-algebra homomorphism $a \to b$.

Now fix a multigraph $\Gamma = (V, E)$ as above, and recall that MMor is the initial algebra of the endofunctor

$$F : X \mapsto V + \mathsf{List}(X) \times_{\mathsf{List}(V)} E$$

of the slice category in Set over $\mathsf{List}(V) \times V$. Let QMMor (for quasi-multimorphisms) be the initial algebra of the endofunctor

$$G : X \mapsto V + \mathsf{Bag}(X) \times_{\mathsf{Bag}(V)} E$$

of the slice category in Set over $\mathsf{List}(V) \times V$. Observe that this slice category also admits a well-defined endofunctor:

$$\overline{F} : X \mapsto V + \mathsf{List}(X) \times_{\mathsf{List}(V)} E$$

Finally, let J be the forgetful from the slice category in Set over $\mathsf{List}(V) \times V$ to that over $\mathsf{Bag}(V) \times V$. We are in the setting of the above propositions with $G_1 = \overline{F}$ and $G_2 = G$. Therefore, there is an induced map:

$$\mathsf{MMor} \to \mathsf{QMMor}$$

This is the same map q that we have seen in different guises above.

Appendix B. Proof of associativity

In this appendix, we prove associativity for the free T-multicategory from Example 4.10.

B.1. **Set-up.** We begin by recalling the set up. Let T be a (finitary) cartesian monad on Set, and let $E \to TV \times V$ be an endomorphism 1-cell in the bicategory $\mathsf{Set}_{(T)}$. The free T-multicategory generated by E has 1-cell given by the initial algebra A of the endofunctor $X \mapsto V + TX \times_{TV} E$ of the category $\mathsf{Set}/TV \times V$. Hence, we have isomorphisms:

$$\mathsf{glue} : V + TA \times_T VE \xrightarrow{\sim} A \qquad\qquad \mathsf{deglue} : A \xrightarrow{\sim} V + TA \times_T VE$$

we also define glue_* to be the restriction of glue to $TA \times_{TV} E$ (which is not an isomorphism). Set:

$$\begin{aligned} A^{(0)} &= V \\ A^{(n)} &= V + T\left(A^{(n-1)}\right) \times_{TV} E \qquad \text{for } n \geq 1 \\ A^{(n)}_* &= T\left(A^{(n-1)}\right) \times_{TV} E \qquad \text{for } n \geq 1 \end{aligned}$$

so that $A^{(n)}_* = A^{(n)} \setminus V$ consists of the "non-constant" elements of $A^{(n)}$. For $n \geq 1$, we have isomorphisms:

$$\begin{aligned} \mathsf{glue}^{(n)}_* : &\qquad T\left(A^{(n-1)}\right) \times_{TV} E \xrightarrow{\sim} A^{(n)}_* \\ \mathsf{deglue}^{(n)}_* : &\qquad A^{(n)}_* \xrightarrow{\sim} T\left(A^{(n-1)}\right) \times_{TV} E \end{aligned}$$

There are the inclusions $\mathsf{inc}_n : A^{(n-1)} \hookrightarrow A^{(n)}$ are defined easily by induction starting with $A^{(0)} = V \hookrightarrow V + E = A^{(1)}$. We have that the initial algebra A is the colimit of the $A^{(n)}$, so that the $A^{(n)}$ form a filtration on A.

B.2. **Composition.** We define a composition 2-cell: $m : TA \times_{TV} A \longrightarrow A$ by induction.

B.2.1. *Base step.* First, the cartesian property of ϵ implies that the map $(\epsilon_A, t_A) : A \to TA \times_{TV} V$ is an isomorphism. We set

$$m^{(0)} : TA \times_{TV} A^{(0)} = TA \times_{TV} V \xrightarrow{\sim} A$$

to be the inverse of this isomorphisms. This completes the base step. It is not strictly necessary (since it will be implied by later constructions), but for illustration purposes we also consider the $n = 1$ case. We have that $A^{(1)} = V + E$, and so:

$$TA \times_{TV} A^{(1)} = TA \times_{TV} V + TA \times_{TV} E$$

we set $m^{(1)}$ to be the composition

$$TA \times_{TV} A^{(1)} \xrightarrow{\sim} TA \times_{TV} V + TA \times_{TV} E \longrightarrow A + A \longrightarrow A$$

where the first map is $m^{(0)} + \mathsf{glue}$ and the second map is the obvious fold.

B.2.2. *Induction step.* For the induction step, suppose we have defined

$$m^{(n-1)} : TA \times_{TV} TA^{(n-1)} \to A$$

for some $n \geq 1$. In order to define the composition map $m^{(n)}$, first observe that, by the definition of $A^{(n)}$ and the fact that products distribute over sums, we have isomorphisms:

$$\begin{aligned} TA \times_{TV} A^{(n)} &\xrightarrow{\sim} TA \times_{TV} \left(V + T(A^{(n-1)}) \times_{TV} E\right) \\ &\xrightarrow{\sim} TA \times_{TV} V + TA \times_{TV} T(A^{(n-1)}) \times_{TV} E \end{aligned}$$

Next, consider the map:

$$A + T\left(TA \times_{TV} A^{(n-1)}\right) \times_{TV} E \longrightarrow TA \times_{TV} V + TA \times_{TV} T(A^{(n-1)}) \times_{TV} E$$

defined as (ϵ_A, t_A) on the frst cofactor and $(\mu_A \circ T\pi_1, T\pi_2) \times 1$ on the second cofactor. The cartesian property of the monad T implies that this map is an isomorphism. Combining the inverse of this isomorphism with the previous observations, we obtain an isomorphism:

$$TA \times_{TV} A^{(n)} \xrightarrow{\sim} A + T\left(TA \times_{TV} A^{(n-1)}\right) \times_{TV} E$$

Finally, we have the composition:

$$A + T(TA \times_{TV} A^{(n-1)}) \times_{TV} E \longrightarrow A + TA \times_{TV} E \longrightarrow A + A \longrightarrow A$$

where the first map is $1 + Tm^{(n-1)} \times 1$, which invokes the induction hypothesis; the second map is $1 + \mathsf{glue}_*$; and the last map is the obvious fold map.

B.2.3. *Observation.* In particular, we have the following commutative diagram:

$$\begin{tikzcd}
T(TA \times_{TV} A^{(n-1)}) \times_{TV} E \arrow[rr, "\simeq"] \arrow[d, "Tm^{(n-1)} \times 1"'] & & TA \times_{TV} TA^{(n-1)} \times_{TV} E \arrow[d, "1 \times \mathsf{glue}_*^{(n)}"] \\
TA \times_{TV} E \arrow[r, "\mathsf{glue}_*"] & A & TA \times_{TV} A_*^{(n)} \arrow[l, "m_*^{(n)}"']
\end{tikzcd}$$

where $m_*^{(n)}$ is $m^{(n)}$ restricted to $A_*^{(n)}$, and the top horizontal map is $(\mu_A \circ T\pi_1, T\pi_2) \times 1$, which is an isomorphism from the fact that T is a cartesian monad. We note that $\mathsf{glue}_*^{(n)}$ is an isomorphism, so that $m_*^{(n)}$ can be defined via this diagram.

B.3. **Associativity.** We are finally in a position to prove the associativity axiom, namely that the following diagram commutes:

$$\begin{tikzcd}
T(TA \times_{TV} A) \times_{TV} A \arrow[rr, "\simeq"] \arrow[d, "Tm \times 1"'] & & TA \times_{TV} TA \times_{TV} A \arrow[d, "1 \times m"] \\
TA \times_{TV} A \arrow[r, "m"] & A & TA \times_{TV} A \arrow[l, "m"']
\end{tikzcd}$$

where, again, the top map is the isomorphism $(\mu_A \circ T\pi_1, T\pi_2) \times 1$. We proceed by induction on the filtration of the last factor of A. The base step is $n = 0$ and follows easily. For the induction step, it suffices to consider $A_*^{(n)}$ instead of all of $A^{(n)}$. Then the desired commutative diagram is:

$$\begin{tikzcd}
T(TA \times_{TV} A) \times_{TV} A_*^{(n)} \arrow[rr, "\simeq"] \arrow[d, "Tm \times 1"'] & & TA \times_{TV} TA \times_{TV} A_*^{(n)} \arrow[d, "1 \times m_*^{(n)}"] \\
TA \times_{TV} A_*^{(n)} \arrow[r, "m_*^{(n)}"] & A & TA \times_{TV} A \arrow[l, "m"']
\end{tikzcd}$$

The proof that this diagram is commutative is illustrated by a sequence of commutative diagrams given in Figure 3. Some explanation of the diagram is in order. For legibility, we omit the subscripts "TV" in the fiber products, and write "1" for the identity maps. All vertical isomorphisms use a map of the form $(\mu \circ T\pi_1, T\pi_2)$, which is an isomorphism due to the cartesian property of the monad T. The top square is commutative since the composition both ways is $Tm \times \mathsf{glue}_*^{(n)}$. We also note that $T(1 \times 1) \times \mathsf{glue}_*^{(n)}$ is an isomorphism. The square below and to the left of the top square commutes due to the cartesian property of the monad T. The pentagon below and to the right of the top square is the diagram of Section B.2.3. The pentagon in the middle of the diagram (ending in $TA \times E$) commutes due to the induction hypothesis. The square below and to the left of the middle pentagon commutes due to the cartesian nature of the monad

T. The pentagon below and to right of the middle pentagon commutes due to the induction hypothesis ($m_*^{(1)} = \mathsf{glue}_*^{(1)}$). Finally the pentagon at the bottom is the diagram in Section B.2.3 above with an extra factor of TA in front, noting that $1 \times 1 \times \mathsf{glue}_*^{(n)}$ is an isomorphism.

$$\begin{tikzcd}
 & & T(TA \times A) \times A_*^{(n)} \arrow[dr, "Tm\times 1"] & & \\
T(TA \times A) \times TA^{(n-1)} \times E \arrow[urr, "T(1\times 1)\times \mathsf{glue}_*^{(n)}"] \arrow[drr, "Tm\times 1\times 1"] & & & TA \times A_*^{(n)} \arrow[dddr, "m_*^{(n)}"] & \\
 & & TA \times TA^{(n-1)} \times E \arrow[ur, "1\times \mathsf{glue}_*^{(n)}"'] & & \\
T(T(TA \times A) \times A^{(n-1)} \times E \arrow[uu, "\simeq"] \arrow[rr, "T(Tm\times 1)\times 1"] & & T(TA \times A^{(n-1)}) \times E \arrow[u, "\simeq"] \arrow[dr, "Tm^{(n-1)}\times 1"] & & \\
 & & & TA \times E \arrow[r, "\mathsf{glue}_*"] & A \\
T(TA \times TA \times A^{(n-1)}) \times E \arrow[uu, "\simeq"] \arrow[rr, "T(1\times m^{(n-1)})\times 1"] \arrow[d, "\simeq"'] & & T(TA \times A) \times E \arrow[ur, "Tm\times 1"'] \arrow[d, "\simeq"'] & & \\
TA \times T(TA \times A^{(n-1)}) \times E \arrow[rr, "1\times Tm^{(n-1)}\times 1"] \arrow[dd, "\simeq"'] & & TA \times TA \times E \arrow[dr, "1\times \mathsf{glue}_*"] & & \\
 & & & TA \times A \arrow[uuur, "m"] & \\
TA \times TA \times TA^{(n-1)} \times E \arrow[rr, "1\times 1\times \mathsf{glue}_*^{(n)}"] & & TA \times TA \times A_*^{(n)} \arrow[ur, "1\times m_*^{(n)}"'] & &
\end{tikzcd}$$

FIGURE 3. Diagram for the induciton step in the proof of associativity of composition in the free T-multicategory defined by a T-graph $E \to TV \times V$. All products are fiber products over TV. Identity maps are denoted by 1.