Polynomial functors and trees

To my Father for his 70th birthday

JOACHIM KOCK

Abstract

We explore the relationship between polynomial functors and (rooted) trees. In the first part we use polynomial functors to derive a new convenient formalism for trees, and obtain a natural and conceptual construction of the category Ω of Moerdijk and Weiss; its main properties are described in terms of some factorisation systems. Although the constructions are motivated and explained in terms of polynomial functors, they all amount to elementary manipulations with finite sets. In the second part we describe polynomial endofunctors and monads as structures built from trees, characterising the images of several nerve functors from polynomial endofunctors and monads into presheaves on categories of trees. Polynomial endofunctors and monads over a base are characterised by a sheaf condition on categories of decorated trees. In the absolute case, one further condition is needed, a certain projectivity condition, which serves also to characterise polynomial endofunctors and monads among (coloured) collections and operads.

Contents

0   Introduction and preliminaries

0.0   Introduction

While linear orders and the category Δ of nonempty finite ordinals constitute the combinatorial foundation for category theory, the theories of operads and multicategories are based on trees. Where Δ is very well understood and admits good formal descriptions, trees are often treated in an ad hoc manner, and arguments about them are often expressed in more or less heuristic terms based on drawings.

Having solid combinatorial foundations is crucial for developing homotopical and higher-dimensional versions of the theories. Recently, Moerdijk and Weiss [17], [18] have undertaken the project of developing a homotopy theory for operads by mimicking the simplicial approach to homotopical category theory. Their work is the main motivation and inspiration for the present article (although no homotopy theory is developed here).

This paper analyses the relationship between polynomial functors, polynomial monads and trees, keeping the analogy with graphs, categories and linear orders as close as possible. In both cases, the interplay between algebra, combinatorics, and homotopy theory follows the same pattern, whose general theory has been worked out by Weber [21]: categories are first defined algebraically, as algebras for a nice monad on the category of graphs; there is a canonical way to distill the combinatorics of such a monad, which in this case yields the category Δ; finally categories are characterised among presheaves on Δ. Analogously, polynomial monads are algebras for a nice monad on the category of polynomial endofunctors; from this monad a category of trees arises naturally; finally polynomial monads are characterised among presheaves on this category of trees.

The theory of polynomial functors is relatively new and has hitherto mostly been explored from the viewpoint of type theory and computer science (some references can be found in [5] or [12]). The point that polynomial functors are an excellent tool for making explicit and analysing the combinatorics underlying operad theory was first made in the paper [13], where polynomial functors were used to extract the first purely combinatorial characterisation of opetopes. Opetopes can be seen as higher-dimensional analogues of trees. The present paper goes to a more fundamental level, substantiating that already the usual notion of tree is of polynomial nature and benefits from this explicitation. The resulting formalism of trees is in fact elementary, has a clear intuitive content, and is easy to work with.

One single observation accounts for the close relationship between polynomial endofunctors and trees, namely that they are represented by diagrams of the same shape, as we now proceed to explain. Although this observation is both natural and fruitful, it seems not to have been made before.

Trees are usually defined and manipulated in either of two ways:

- 'Topological/static characterisation': trees are graphs $E \rightrightarrows V$ with certain topological properties and structure (a base point).

- 'Recursive characterisation': a tree is either a trivial tree or a collection of smaller trees.

In this work, a different viewpoint is taken, specifically designed for the use of trees in operad theory and related topics:

- 'Operational characterisation': trees are certain many-in/one-out structures, i.e. built from building blocks like

in

out
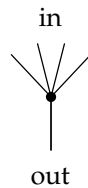
Accordingly a tree should have a set of edges A, and a set of vertices (or nodes) N, which we think of as operations; these should have inputs and output (source and target). So the structure is something like

$$N$$

in   out

$$A \qquad\qquad A.$$
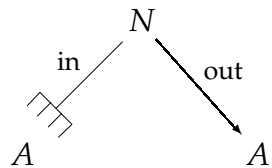
The fork represents a 'multi-valued map', because a node may have several input edges. A standard way to encode multi-valued maps is in terms of correspondences or spans; hence we arrive at this shape of diagram to represent a tree:

$$A \xleftarrow{\;s\;} M \xrightarrow{\;p\;} N \xrightarrow{\;t\;} A; \tag{1}$$

to be explicit, M is the set of all input edges (i.e. pairs (b, e) where b is a node and e is an input edge of b). In order to be trees, such diagrams should satisfy certain axioms, which turn out to be quite simple.

Although this is clearly also a static graph-like definition, its operational aspect comes to the fore with the observation that this shape of diagram is precisely what defines polynomial endofunctors [12], vindicating the interpretation of N as a set of operations. The polynomial endofunctor represented by a diagram (1) is

$$\mathbf{Set}/A \xrightarrow{\;s^*\;} \mathbf{Set}/M \xrightarrow{\;p_*\;} \mathbf{Set}/N \xrightarrow{\;t_!\;} \mathbf{Set}/A.$$

Among all polynomial endofunctors we characterise those that correspond to trees, and since the involved conditions have a clear intuitive content and are convenient to work with, we will simply take this as the *definition* of tree (1.0.3).

If in (1) the map p is the identity, the diagram is just that of a directed (non-reflexive) graph, and the associated polynomial functor becomes *linear*. Imposing the tree axioms

in this case yields linear trees, i.e. finite linear orders. Linear polynomial monads are the same thing as small categories.

Polynomial endofunctors and monads are intimately related with collections and operads (the precise relationship is given in Sections 2.4 and 2.5), but are distinguished by a representability feature: they can be represented by diagrams of sets like (1), and most constructions with polynomial functors can be performed in terms of elementary operations on those representing sets. As a result, all operations on trees can be carried out completely formally (e.g., grafting is given in terms of pushouts of finite sets (1.1.19)), without ever having seen a tree in nature — although of course the arguments are easier to follow with drawings of trees in mind.

The recursive aspect of trees is also prominent in the present approach, remembering that polynomial endofunctors provide categorical semantics for inductive data types (*W*-types), the latter appearing as initial Lambek algebras for the former [16]. In fact, a recursive characterisation of trees (1.1.21) follows quite easily from the definition. While in type theory trees (of a certain branching profile P) appear as initial algebras *for* some polynomial functor P (expressing the branching profile), in this work trees *are* themselves certain polynomial functors. In a precise sense they are absolute trees, i.e. not relative to any preassigned branching profile.

The paper naturally divides into two parts: the first part concerns the categories of trees. Most arguments in this part are quite elementary, and some of the initial manœuvres may appear pedantic. They are deliberately included in order to emphasise the workability of the new tree formalism — the reader is challenged to provide easier arguments in other formalisms of trees. The second part uses the tree formalism to prove theorems about polynomial functors and polynomial monads and to clarify the relationship with operads. This part is of a more technical nature and requires some more category theory.

We proceed to give an overview of each of the two parts of the paper.

0.0.1. **Overview of Part 1: trees in terms of polynomial endofunctors.** After recalling the relevant notions from the theory of polynomial functors, we define a tree to be a diagram of sets of shape (1) satisfying four simple conditions (1.0.3). The category **TEmb** is the full subcategory of **PolyEnd** (the category of polynomial endofunctors) consisting of the trees. The morphisms are diagrams

$$
\begin{array}{ccccccc}
A' & \longleftarrow & M' & \longrightarrow & N' & \longrightarrow & A' \\
\downarrow & & \downarrow & \lrcorner & \downarrow & & \downarrow \\
A & \longleftarrow & M & \longrightarrow & N & \longrightarrow & A.
\end{array}
$$

The symbol **TEmb** stands for 'tree embeddings', as it turns out maps between trees are always injective (1.1.3) and correspond to a notion of subtree. Root-preserving embeddings and ideal embeddings are characterised in terms of pullback conditions, and every tree embedding factors as root-preserving followed by ideal embedding (1.1.15). These two classes of maps allow pushouts along each other in the category **TEmb** —

this is grafting of trees (1.1.19). This leads to a recursive characterisation of trees (1.1.21), as well as the useful result that every tree is the iterated pushout of its one-node subtrees over its inner edges (1.1.24).

For a polynomial endofunctor P, a P-tree is a tree with a map to P. This amounts to structuring the set of input edges of each node. For example, if M is the free-monoid monad (0.1.8), then M-trees are precisely planar trees. It is shown, using the recursive characterisation of trees, that the set of isomorphism classes of P-trees, denoted tr(P), is the least fixpoint (initial Lambek algebra) for the polynomial endofunctor $1 + P$ (Theorem 1.2.5). This leads to the following explicit construction of the free monad on P: if P is given by the diagram $A \leftarrow M \rightarrow N \rightarrow A$, then the free monad on P is given by

$$A \leftarrow \mathrm{tr}'(\mathsf{P}) \rightarrow \mathrm{tr}(\mathsf{P}) \rightarrow A$$

where $\mathrm{tr}'(\mathsf{P})$ denotes the set of isomorphism classes of P-trees with a marked leaf (1.2.8). The monad structure is given by grafting P-trees. We are particularly interested in free monads on trees. Since maps between trees are embeddings, the free monad on a tree $\mathsf{T} = (A \leftarrow M \rightarrow N \rightarrow A)$ is given by

$$A \leftarrow \mathrm{sub}'(\mathsf{T}) \rightarrow \mathrm{sub}(\mathsf{T}) \rightarrow A$$

(where $\mathrm{sub}(\mathsf{T})$ (resp. $\mathrm{sub}'(\mathsf{T})$) denotes the set of subtrees of T (resp. subtrees with a marked leaf)).

We now define **Tree** to be the category whose objects are trees and whose arrows are maps between their free monads (1.3.1). In other words, **Tree** is a full subcategory of **PolyMnd** (the category of polynomial monads): it is the Kleisli category of **TEmb** with respect to the free-monad monad on **PolyEnd**. (It is shown that any map in **PolyEnd** between free monads on trees is a monad map (1.3.2).) In explicit terms, morphisms send edges to edges and subtrees to subtrees. The category **Tree** is equivalent to the category Ω of Moerdijk and Weiss [17], whose presheaves are called dendroidal sets. Its construction in terms of polynomial functors reveals important properties analogous to properties of Δ. In fact, Δ is equivalent to the full subcategory in **Tree** consisting of the linear trees.

The main intrinsic features of the category **Tree** are expressed in terms of three factorisation systems: **Tree** is shown to have has surjective/injective factorisation, generic/free factorisation, as well as root-preserving/ideal-embedding factorisation. The generic maps are precisely the boundary-preserving maps, and the generic/free factorisation system plays an important role in the second part of the paper. The compatibilities between the three factorisation systems are summarised in this figure:

surjective	injective	
generic	free	
root preserving	ideal	
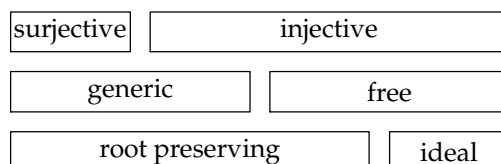
As suggested by the figure, every arrow factors essentially uniquely as a surjection followed by a generic injection, followed by a free root-preserving map followed by

an ideal embedding. Explicit descriptions are derived for each of these four classes of maps. The surjections consist in deleting unary nodes, the generic injections are node refinements (and of course the free maps are the tree embeddings). The ideal embeddings are those corresponding to subtrees containing all descendants — this is the notion of subtree most relevant to computer science and linguistics.

The subcategory of generic tree maps is opposite to the category of trees studied by Leinster [14, §7.3]. On Leinster's side of the duality, tree maps can be described in terms of set maps between the sets of nodes. On our side of the duality, tree maps are described in terms of set maps between the sets of edges. The category of generic injections is roughly the opposite of the category of trees studied[1] by Ginzburg and Kapranov in their seminal paper [7]; the difference is that they exclude all trees with nullary nodes. In fact, most of the time they also exclude trees with unary nodes, and call the remaining trees reduced.

The subcategory **TEmb**, orthogonal to the generics, does not seem to have been studied before. It allows grafting of trees to be expressed as a pushout (1.1.19) and it carries the Grothendieck topology in terms of which the Segal condition is expressed (2.3.3).

0.0.2. **Overview of Part 2: polynomial endofunctors in terms of trees.** In the second part we describe polynomial endofunctors and monads as structures built from trees. Let **tEmb** and **tree** denote chosen skeleta of **TEmb** and **Tree**, respectively. Since **tEmb** is a subcategory in **PolyEnd**, there is a natural nerve functor **PolyEnd** → **PrSh**(**tEmb**), and similarly there is a nerve functor **PolyMnd** → **PrSh**(**tree**). These nerve functors are fully faithful, and we characterise their images. A main tool for these results is the theory of monads with arities due to Weber [21], which is reviewed in Section 2.0. Nerves of polynomial functors constitute an interesting application of Weber's theory, of a somewhat different flavour than the previously known examples, the new twist being that **PolyEnd** is not a presheaf category.

A key observation is that although **PolyEnd** itself is not a presheaf category, every slice of it *is* a presheaf category. This result relies on a notion of element of a polynomial endofunctor, introduced in Section 2.1: the elements of a polynomial endofunctor are the maps into it from elementary trees, i.e. trees with at most one node. The elementary trees, forming the subcategory **elTr**, play the role of representables: we show (2.1.3) that the slice category **PolyEnd**/P is naturally equivalent to the presheaf category **PrSh**(el(P)), and that each polynomial endofunctor P is the colimit of a diagram of shape el(P) (cf. 2.1.7).

In Section 2.2 we come to the notions of generic morphism and generic factorisation, key notions in Weber's theory. We show that every element of a polynomial monad factors as generic followed by free, and the object appearing in the middle of the factorisation is a tree (2.2.4). This is a main ingredient in the proof that the free-monad monad on **PolyEnd** is a local right adjoint (2.2.6). We then show that trees

---

[1]It seems that the category they study is not the same as the category they define: their definition 1.1.4 does not seem to exclude contraction of external edges. I mention this curiosity as an illustration of the subtlety of formalising arguments with trees.

provide arities for the free-monad monad on **PolyEnd** (cf. 2.2.10). With these facts established, Weber's general nerve theorem (2.0.5) implies the following characterisation (2.2.12): a presheaf on **tree** is a polynomial monad if and only if its restriction to **tEmb** is a polynomial endofunctor.

What is here called the special nerve theorem (2.0.3), first proved by Leinster [15] and subsumed in the theory of Weber [21], concerns the case of a local right adjoint cartesian monad on a presheaf category; it characterises nerves in terms of the Segal condition. The Segal condition is about requiring certain canonical cocones to be sent to limit cones, and can also be formulated as a sheaf condition for a Grothendieck topology. The Segal condition makes sense also in the present case, and amounts to a sheaf condition on **tEmb** (2.3.3). It is shown that the nerve of a polynomial endofunctor is always a sheaf (2.3.4), and we have an equivalence of categories $\mathbf{Sh}(\mathbf{tEmb}) \simeq \mathbf{PrSh}(\mathbf{elTr})$. However, the Segal condition is not enough to characterise nerves of polynomial monads. The special nerve theorem does apply to slices, though, (they are presheaf categories): for a fixed polynomial monad P, monads over P are characterised (2.3.6) as presheaves on **tree**/P satisfying the Segal condition.

In the absolute case, one more condition is needed for a nerve theorem: it amounts to characterising the polynomial endofunctors among the presheaves on **elTr**. Presheaves on **elTr** are precisely (coloured) collections. We show that a collection is (isomorphic to) the nerve of a polynomial endofunctor if and only if it is (isomorphic to) the symmetrisation of a nonsymmetric collection (2.4.5). (More precisely, the category of polynomial endofunctors is the Kleisli category for the symmetrisation monad for nonsymmetric collections). Another characterisation is also obtained: the polynomial endofunctors are the projective objects in **Coll** with respect to colour-preserving surjections (2.4.8). Such collections are called flat.

The final section contains a big diagram relating the various objects involved: polynomial endofunctors and monads (as well as their planar versions), collections and operads (as well as the nonsymmetric versions), and the adjunctions and nerve functors relating them. The nerve functor for polynomial monads factors through the category of coloured operads. Polynomial monads are characterised as those operads whose underlying collection is flat.

A note about generality. In this paper, for simplicity, we only consider finite trees, and correspondingly we always assume our polynomial functors to be finitary. This is the natural level of generality from the viewpoint of operad theory, and for the sake of giving a formal construction of the category of trees of Moerdijk and Weiss, which was the original motivation for this work. Most results and constructions should generalise to wellfounded trees (in the category of sets) and arbitrary polynomial functors. I believe large parts of the theory will also generalise to an arbitrary locally cartesian closed category $\mathscr{E}$. Many proofs can be reinterpreted in the internal language of $\mathscr{E}$, but there are also some that cannot (e.g. involving complements), and new approaches may be required. This general case, perhaps more interesting from the viewpoint of type theory, is left to another occasion.

The ideas and techniques of this paper are currently being developed in another

direction (in joint work with André Joyal) to account also for graphs. We introduce a new formalism for Feynman graphs and prove nerve theorems for cyclic and modular operads [10].

0.1 Polynomial functors

0.1.1. **Notation.** Throughout we denote by 0 the empty set and by 1 the singleton. We use the symbols $+$ and $\sum$ for disjoint union of sets (or categories).

We recall some facts about polynomial functors. For further details and many other aspects of this fascinating topic, the reader is referred to the manuscript in preparation *Notes on Polynomial Functors* [12].

0.1.2. **Polynomial functors.** A diagram of sets and set maps like this

$$
\begin{array}{ccc}
E & \xrightarrow{\ p\ } & B \\
{}^{s}\swarrow & & \searrow{}^{t} \\
I & & J
\end{array}
\tag{2}
$$

gives rise to a *polynomial functor* $\mathrm{P} : \mathbf{Set}/I \to \mathbf{Set}/J$ defined by

$$
\mathbf{Set}/I \xrightarrow{\ s^{*}\ } \mathbf{Set}/E \xrightarrow{\ p_{*}\ } \mathbf{Set}/B \xrightarrow{\ t_{!}\ } \mathbf{Set}/J.
$$

Here lowerstar and lowershriek denote, respectively, the right adjoint and the left adjoint of the pullback functor upperstar. In explicit terms, the functor is given by

$$
\begin{array}{ccc}
\mathbf{Set}/I & \longrightarrow & \mathbf{Set}/J \\
[f : X \to I] & \longmapsto & \displaystyle\sum_{b \in B} \prod_{e \in E_{b}} X_{s(e)}
\end{array}
\tag{3}
$$

where $E_{b} := p^{-1}(b)$ and $X_{i} := f^{-1}(i)$, and where the last set is considered to be over J via $t_{!}$.

In this paper we shall only consider polynomial functors for which the map p has finite fibres (equivalently, the functor preserves sequential colimits.) Such polynomial functors are called finitary. From now on, 'polynomial functor' means 'finitary polynomial functor'.

0.1.3. **Categories of polynomial functors.** (Cf. [6].) There is a category $\mathbf{Poly}(I, J)$ whose objects are the polynomial functors from $\mathbf{Set}/I$ to $\mathbf{Set}/J$, and whose arrows are the cartesian natural transformations (i.e. natural transformations with cartesian naturality squares). A cartesian natural transformation $u : \mathrm{P}' \Rightarrow \mathrm{P}$ between polynomial functors

corresponds precisely to a commutative diagram

$$
\begin{array}{ccc}
E' & \xrightarrow{\;p'\;} & B' \\
\end{array}
\tag{4}
$$

whose middle square is cartesian. In other words, giving u amounts to giving a J-map $u : B' \to B$ together with an I-bijection $E'_{b'} \xrightarrow{\sim} E_{u(b')}$ for each $b' \in B'$.

The composition of two polynomial functors is again polynomial [6]; this is a consequence of distributivity and the Beck-Chevalley conditions. Clearly the identity functor of $\mathbf{Set}/I$ is polynomial for each I. It follows that the categories $\mathbf{Poly}(I, J)$ form the hom categories of a 2-category $\mathbf{Poly}$, which we see as a sub-2-category of $\mathbf{Cat}$: the objects are the slice categories $\mathbf{Set}/I$, the arrows are the polynomial functors, and the 2-cells are the cartesian natural transformations. Since everything sits inside $\mathbf{Cat}$, associativity of the compositions as well as the interchange law for composition of 2-cells are automatic.

From now on we shall only be concerned with the case $J = I$, i.e. the case of endofunctors. Throughout we use sans serif typeface for polynomial endofunctors, writing $\mathsf{P} = (P^0, P^1, P^2)$ for the functor represented by

$$
P^0 \xleftarrow{\;s\;} P^2 \xrightarrow{\;p\;} P^1 \xrightarrow{\;t\;} P^0.
$$

We shall use the letters s, p, t for the three arrows in any diagram representing a polynomial endofunctor.

0.1.4. **Polynomial monads.** A *polynomial monad* is a monad in the 2-category $\mathbf{Poly}$, i.e. a polynomial endofunctor $\mathsf{P} : \mathbf{Set}/I \to \mathbf{Set}/I$ with monoid structure in the monoidal category $(\mathbf{PolyEnd}(I), \circ, \mathsf{Id})$. More explicitly still, there is specified a composition law $\mu : \mathsf{P} \circ \mathsf{P} \Rightarrow \mathsf{P}$ with unit $\eta : \mathsf{Id} \Rightarrow \mathsf{P}$, satisfying the usual associativity and unit conditions, and μ and η are cartesian natural transformations. Let $\mathbf{PolyMnd}(I)$ denote the category of polynomial monads on $\mathbf{Set}/I$. The arrows are cartesian natural transformations respecting the monad structure.

0.1.5 Proposition. *(cf. [5], [6].) The forgetful functor $\mathbf{PolyMnd}(I) \to \mathbf{PolyEnd}(I)$ has a left adjoint, denoted $\mathsf{P} \mapsto \overline{\mathsf{P}}$. The monad $\overline{\mathsf{P}}$ is the* free monad *on P.*

An explicit construction of $\overline{\mathsf{P}}$ is given in 1.2.7.

0.1.6. **Variable endpoints.** It is necessary to consider also 2-cells between polynomial functors with different endpoints. Let $\mathbf{PolyEnd}$ denote the category whose objects are

polynomial functors $\mathsf{P} = (P^0 \leftarrow P^2 \rightarrow P^1 \rightarrow P^0)$ and whose morphisms are diagrams

$$
\begin{array}{ccccccc}
Q^0 & \longleftarrow & Q^2 & \longrightarrow & Q^1 & \longrightarrow & Q^0 \\
\downarrow{\scriptstyle \alpha^0} & & \downarrow{\scriptstyle \alpha^2} \;\lrcorner & & \downarrow{\scriptstyle \alpha^1} & & \downarrow{\scriptstyle \alpha^0} \\
P^0 & \longleftarrow & P^2 & \longrightarrow & P^1 & \longrightarrow & P^0.
\end{array}
\tag{5}
$$

This category is fibred over **Set** by returning the endpoint [6].

A morphism α of polynomial functors is called *injective*, resp. *surjective*, if each of the three components, $\alpha^0, \alpha^1, \alpha^2$ is injective, resp. surjective.

Let **PolyMnd** denote the category whose objects are polynomial monads and whose morphisms are diagrams like (5), required to respect the monad structure. All we need to know about this is:

0.1.7 Proposition. *(Cf. [6].) The forgetful functor* **PolyMnd** $\rightarrow$ **PolyEnd** *has a left adjoint* $\mathsf{P} \mapsto \overline{\mathsf{P}}$, *the free-monad functor. In other words, for each polynomial endofunctor* P *and each polynomial monad* M, *there is a bijection*

$$\textbf{PolyEnd}(\mathsf{P}, \mathsf{M}) \leftrightarrow \textbf{PolyMnd}(\overline{\mathsf{P}}, \mathsf{M}),$$

natural in P *and* M.

This adjunction restricts to the adjunction of 0.1.5 in each fibre. It is not a fibred adjunction, though.

0.1.8. **Examples.** The free-monoid monad

$$
\begin{array}{rcl}
\mathsf{M} : \textbf{Set} & \longrightarrow & \textbf{Set} \\
X & \longmapsto & \displaystyle\sum_{n \in \mathbb{N}} X^n
\end{array}
$$

is polynomial: it is represented by the diagram

$$1 \longleftarrow \mathbb{N}' \longrightarrow \mathbb{N} \longrightarrow 1,$$

where $\mathbb{N}' \rightarrow \mathbb{N}$ is such that the fibre over n has cardinality n, like for example $\mathbb{N}' := \{(i, n) \in \mathbb{N} \times \mathbb{N} \mid i < n\}$ with the second projection. The slice category **PolyMnd**$/\mathsf{M}$ of polynomial monads over M is equivalent to the category of small multicategories (also called nonsymmetric coloured operads), and the fibre **PolyMnd**$(1)/\mathsf{M}$ corresponds to nonsymmetric operads.

The identity functor $\mathsf{Id} : \textbf{Set} \rightarrow \textbf{Set}$ is clearly polynomial. The slice category **PolyMnd**$/\mathsf{Id}$ is equivalent to the category of small categories, and the fibre **PolyMnd**$(1)/\mathsf{Id}$ is equivalent to the category of monoids.

More generally, polynomial endofunctors over a polynomial monad T correspond to T-graphs, and polynomial monads over T correspond to small T-multicategories. All these results can be found in Leinster's book [14, §4.2], modulo the observation that any endofunctor with a cartesian natural transformation to a polynomial one is again polynomial, cf. [12].

1   Trees in terms of polynomial endofunctors

1.0   Trees

We shall define trees to be certain polynomial endofunctors. To motivate this definition, let us first informally explain what trees are supposed to be, and then show how to associate a polynomial endofunctor to a tree.

1.0.1. **Trees.** Our trees are non-planar finite rooted trees with boundary. Each node has a finite number of incoming edges and precisely one outgoing edge, always drawn below the node. The following drawings should suffice to exemplify trees, but beware that the planar aspect inherent in a drawing should be disregarded:
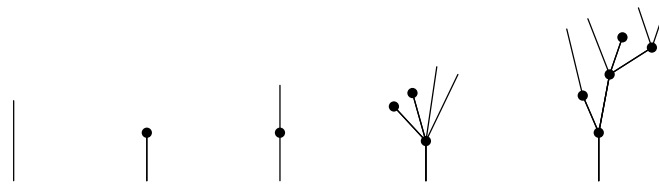
Note that certain edges (the *leaves*) do not start in a node and that one edge (the *root edge*) does not end in a node. The leaves and the root together form the *boundary* of the tree.

We shall give a formal definition of tree in a moment (1.0.3).

1.0.2. **Polynomial functors from trees.** Given a tree, define a polynomial functor

$$T^0 \xleftarrow{\ s\ } T^2 \xrightarrow{\ p\ } T^1 \xrightarrow{\ t\ } T^0,$$

by letting T^0 be the set of edges, T^1 the set of nodes, and T^2 the set of nodes with a marked input edge, i.e. the set of pairs (b, e) where b is a node and e is an incoming edge of b. The maps are the obvious ones: s returns the marked edge of the node (i.e. $(b, e) \mapsto e$), the map p forgets the mark (i.e. $(b, e) \mapsto b$), and t returns the output edge of the node.

For example, the first three trees in the drawing above correspond to the following polynomial functors:

$$1 \leftarrow 0 \rightarrow 0 \rightarrow 1 \qquad\qquad 1 \leftarrow 0 \rightarrow 1 \rightarrow 1 \qquad\qquad 2 \leftarrow 1 \rightarrow 1 \rightarrow 2.$$

The polynomial functors that arise from this construction are characterised by four simple conditions which are convenient to work with. We shall take this characterisation as our definition of tree:

1.0.3. **Definition of tree.** We define a *finite rooted tree with boundary* to be a polynomial endofunctor $\mathsf{T} = (T^0, T^1, T^2)$

$$T^0 \xleftarrow{\ s\ } T^2 \xrightarrow{\ p\ } T^1 \xrightarrow{\ t\ } T^0$$

satisfying the following four conditions:
    (1) all the involved sets are finite.
    (2) t is injective
    (3) s is injective with singleton complement (called the *root* and denoted 1).
With $T^0 = 1 + T^2$, define the walk-to-the-root function $\sigma : T^0 \to T^0$ by $1 \mapsto 1$ and
$e \mapsto t(p(e))$ for $e \in T^2$.
    (4) $\forall x \in T^0 : \exists k \in \mathbb{N} : \sigma^k(x) = 1$.

The elements of T^0 are called *edges*. The elements of T^1 are called *nodes*. For $b \in T^1$,
the edge $t(b)$ is called the *output edge* of the node. That t is injective is just to say that
each edge is the output edge of at most one node. For $b \in T^1$, the elements of the fibre
$(T^2)_b := p^{-1}(b)$ are called *input edges* of b. Hence the whole set $T^2 = \sum_{b \in T^1} (T^2)_b$ can
be thought of as the set of nodes-with-a-marked-input-edge, i.e. pairs (b, e) where b is
a node and e is an input edge of b. The map s returns the marked edge. Condition (3)
says that every edge is the input edge of a unique node, except the root edge. Condition
(4) says that if you walk towards the root, in a finite number of steps you arrive there.

The edges not in the image of t are called *leaves*. The root and the leaves together
form the *boundary* of the tree.

From now on we just say *tree* for 'finite rooted tree with boundary'.

Let us briefly describe how to draw such a tree, i.e. give the converse of the con-
struction in 1.0.2. Given (T^0, T^1, T^2) we define a finite, oriented graph with boundary,
i.e. edges are allowed to have loose ends: take the vertex set to be T^1 and the edge set
to be T^0. The edges $x \in T^0$ which are not in the image of t are the input edges of the
graph in the sense that they do not start in a vertex. For each other edge x, we let it start
in b if and only if $t(b) = x$. (Precisely one such b exists by axiom (2).) Clearly every
b occurs like this. Now we have decided where each edge starts. Let us decide where
they end: the root edge 1 is defined to be the output edge of the graph, in the sense that
it does not end in a vertex. For each other edge $e \neq 1$ (which we think of as $e \in T^2$), we
let it end in $p(e)$. Note that the fibre of p over a vertex b consists of precisely the edges
ending in b. Now we have described how all the edges and vertices are connected, and
hence we have described a finite, oriented graph with boundary. Condition (4) implies
that the graph is connected: every $e \neq 1$ has a 'next edge' $\sigma(e)$ distinct from itself, and
in a finite number of steps comes down to the root edge. There can be no loops because
there is precisely one edge coming out of each vertex, and linear cycles are excluded
by connectedness and existence of a root. In conclusion, the graph we have drawn is a
tree.

1.0.4. **The trivial tree.** The nodeless tree

$$1 \longleftarrow 0 \longrightarrow 0 \longrightarrow 1,$$

(consisting solely of one edge) is called the *trivial tree*, and is denoted ı.

1.0.5. One-node trees. For each finite set E we have a *one-node tree*,

$$E + 1 \xleftarrow{\;\;s\;\;} E \xrightarrow{\;\;p\;\;} 1 \xrightarrow{\;\;t\;\;} E + 1,$$

where s and t are the sum inclusions.

1.0.6. Elementary trees. An *elementary tree* is one with at most one node. That is, either a trivial tree or a one-node tree. These will play a fundamental role in the theory. We shall see in a moment that every tree is obtained by gluing together one-node trees along trivial trees in a specific way (grafting), while polynomial endofunctors are more general colimits of elementary trees.

1.0.7. Terminology. We define a partial order (called the *tree order*) on the edge set T^0 by declaring $x \leq y$ when $\exists k \in \mathbb{N} : \sigma^k(x) = y$. In this case x is called a *descendant* of y, and y is called an *ancestor* of x. In the particular case where $\sigma(x) = y$ and $x \neq y$, we say that x is a *child* of y. If $\sigma(x) = \sigma(y)$ and $x \neq y$ we say that x and y are *siblings*. We define the *distance* from x to y to be $\min\{k \in \mathbb{N} \mid \sigma^k(x) = y\}$, whenever this set is nonempty. Note that the order induced on any 'up-set' is a linear order: if $e \leq x$ and $e \leq y$ then $x \leq y$ or $y \leq x$. The poset T^0 has a maximal element, namely the root; hence it has binary joins: the join of two edges is their nearest common ancestor. Every leaf is a minimal element for the tree order, but there may be other minimal elements. (Note that a partial order is induced on $T^2 \subset T^0$, and also on T^1 (via t).)

1.1   The category *TEmb*

1.1.1. The category of trees and tree embeddings. Define the category *TEmb* to be the full subcategory of *PolyEnd* consisting of the trees. Hence a map of trees $\phi : \mathsf{S} \to \mathsf{T}$ is a diagram

$$
\begin{array}{ccccccc}
S^0 & \longleftarrow & S^2 & \longrightarrow & S^1 & \longrightarrow & S^0 \\
\downarrow{\scriptstyle \phi^0} & & \downarrow{\scriptstyle \phi^2}\,\lrcorner & & \downarrow{\scriptstyle \phi^1} & & \downarrow{\scriptstyle \phi^0} \\
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0
\end{array}
\tag{6}
$$

The cartesian condition amounts to 'arity preservation': the set of input edges of $b \in S^1$ maps bijectively onto the set of input edges of $\phi^1(b)$. Root and leaves are not in general preserved.

1.1.2 Lemma. *Morphisms in **TEmb** preserve the childhood relation. That is, for a morphism $\phi : \mathsf{S} \to \mathsf{T}$, if x is a child edge of y in S then $\phi^0(x)$ is a child edge of $\phi^0(y)$ in T. More generally, morphisms preserve distance.*

Proof. To say that x is a child of y means that x is not the root and $t(p(x)) = y$. The property of not being the root is preserved by any map (cf. commutativity of the left-hand square in the diagram), so $\phi^0(x)$ is not the root either. Now apply ϕ and use that it commutes with p and t, cf. (6). $\qquad\qquad\square$

1.1.3 Proposition. *Every morphism in **TEmb** is injective.*

Proof. Let $\phi : S \to T$ in **TEmb**. Let $r \in T^0$ denote the image of the root edge. Let x, y be edges in S and suppose $\phi^0(x) = \phi^0(y)$. Since ϕ^0 preserves distance we have $d(x, 1) = d(\phi^0(x), r) = d(\phi^0(y), r) = d(y, 1)$. Since x and y have the same distance to the root, it makes sense to put $k := \min\{n \in \mathbb{N} \mid \sigma^n(x) = \sigma^n(y)\}$, and $z := \sigma^k(x) = \sigma^k(y)$ (nearest common ancestor). If $k > 0$, then the edges $\sigma^{k-1}(x)$ and $\sigma^{k-1}(y)$ are both children of z, and by childhood preservation, we have $\phi(\sigma^{k-1}(x)) = \phi(\sigma^{k-1}(y))$. But ϕ induces a bijection between the fibre $(S^2)_z$ and the fibre $(T^2)_{\phi^0(z)}$, so we conclude that already $\sigma^{k-1}(x) = \sigma^{k-1}(y)$, contradicting the minimality of k. Hence $k = 0$, which is to say that already $x = y$. Hence we have shown that ϕ^0 is injective. Since t is always injective, it follows that also ϕ^1 and ϕ^2 are injective. □

The proposition shows that the category **TEmb** is largely concerned with the combinatorics of subtrees, which we now pursue. It must be noted, though, that the category contains nontrivial automorphisms. In particular it is easy to see that

1.1.4 Lemma. *The assignment of a one-node tree to every finite set as in 1.0.5 defines a fully faithful functor from the groupoid of finite sets and bijections into **TEmb**. (The essential image consists of the trees with precisely one node.)* □

1.1.5. **Subtrees.** A subtree of a tree T is an isomorphism class of arrows $S \to T$ in **TEmb**; more concretely it is an arrow $S \to T$ for which each of the three set maps are subset inclusions. Translating into classical viewpoints on trees, subtree means connected subgraph with the property that if a node is in the subgraph then all its incident edges are in the subgraph too.

Here are two examples:
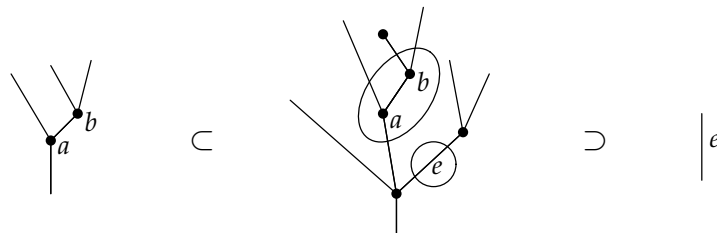
1.1.6. **Edges.** For each edge x of T there is a subtree $\iota \to T$ given by

$$
\begin{array}{ccccccc}
1 & \longleftarrow & 0 & \longrightarrow & 0 & \longrightarrow & 1 \\
{\scriptstyle \ulcorner x \urcorner} \downarrow & & \downarrow & & \downarrow & & \downarrow {\scriptstyle \ulcorner x \urcorner} \\
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0.
\end{array}
$$

The subtree consists solely of the edge x. The edge is the root edge iff the left-hand square is a pullback, and the edge is a leaf iff the right-hand square is a pullback.

1.1.7. **One-node subtrees.** For each node b in T there is a subtree inclusion

$$(T^2)_b + 1 \longleftarrow (T^2)_b \longrightarrow \{b\} \longrightarrow (T^2)_b + 1$$
$$\downarrow \qquad\qquad \downarrow \qquad \llcorner \quad \downarrow \qquad\qquad \downarrow$$
$$T^0 \longleftarrow T^2 \longrightarrow T^1 \longrightarrow T^0$$

The vertical maps at the ends are the sum of $s \mid (T^2)_b$ and the map sending 1 to $t(b)$. The subtree defined is the local one-node subtree at b: the node itself with all its incident edges.

1.1.8 Proposition. *Let* R *and* S *be nontrivial subtrees in* T*, and suppose that* $R^1 \subset S^1$*. Then* R $\subset$ S*. In particular, a nontrivial subtree is determined by its nodes.*

Proof. We need to provide the dotted arrows in the diagram

$$R^0 \longleftarrow R^2 \longrightarrow R^1 \longrightarrow R^0$$
$$S^0 \longleftarrow S^2 \longrightarrow S^1 \longrightarrow S^0$$
$$T^0 \longleftarrow T^2 \longrightarrow T^1 \longrightarrow T^0$$

The arrow $R^1 \to S^1$ is the assumed inclusion of nodes. For each node b in R we have a bijection between the fibre $(R^2)_b$ and the fibre $(S^2)_b$. These bijections assemble into a map $R^2 \to S^2$ and a cartesian square. Since $R^0 = R^2 + \{r\}$ where r is the root edge of R, to specify the arrow $R^0 \to S^0$ it remains to observe that r maps into S^0: indeed, there is a $b \in R^1$ with $t(b) = r$. Hence $\phi^0(r) = \phi^0(t(b)) = t(\phi^1(b)) \in S^0$. $\square$

1.1.9. **Ideal subtrees.** An *ideal subtree* is a subtree containing all the descendant nodes of its edges, and hence also all the descendant edges. (Hence it is a 'down-set' for the tree order (both with respect to nodes and with respect to edges), and just by being a subtree it is also closed under binary join.)

Each edge z of a tree T determines an ideal subtree denoted D_z:

$$D_z : \quad D^0 \longleftarrow D^2 \longrightarrow D^1 \longrightarrow D^0$$
$$\qquad T^0 \longleftarrow T^2 \longrightarrow T^1 \longrightarrow T^0$$

where

$$
\begin{aligned}
D^0 &:= \{x \in T^0 \mid x \le z\}, \\
D^1 &:= \{b \in T^1 \mid t(b) \in D^0\}, \\
D^2 &:= \{e \in T^2 \mid t(p(e)) \in D^0\} = D^0 \smallsetminus \{z\}.
\end{aligned}
$$

It is easy to check that this is a tree; it looks like this:
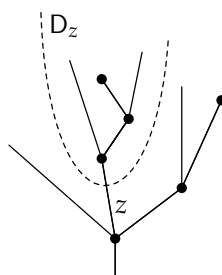
Note also that we have $x \in D_z \Leftrightarrow x \leq z$.

1.1.10 Lemma.*The following are equivalent for a tree embedding $\phi : S \to T$:*

    1.The image subtree is an ideal subtree.

    2.The right-hand square is cartesian (like in the above diagram).

    3.The image of each leaf is again a leaf.

Proof. (1) $\Rightarrow$ (2): clearly every ideal subtree $S \subset T$ is equal to D_z for z the root of S. Hence the embedding has cartesian right-hand square.

(2) $\Rightarrow$ (3): a leaf in S is characterised (1.1.6) as an edge for which the right-hand square is cartesian; composing with ϕ gives then again a cartesian right-hand square, so the edge is again a leaf in T.

(3) $\Rightarrow$ (1): let x be an edge in S, having a child node b in T (that is, $p(b) = x$). This means x is not a leaf in T, and hence by assumption, not a leaf in S either. So b is also in S. $\qquad\square$

1.1.11. **Pruning.** Using complements, it is not difficult to see that an edge $z \in T^0$ defines also another subtree which has the original root, but where all descendants of z have been pruned. In other words, the ideal subtree D_z is thrown away (except for the edge z itself). Formally, with the notation of the ideal subtree: put $C^1 := T^1 \smallsetminus D^1$ and $C^2 := T^2 \smallsetminus D^2$. Then clearly we have a cartesian square

$$
\begin{array}{ccc}
C^2 & \longrightarrow & C^1 \\
\downarrow & \lrcorner & \downarrow \\
T^2 & \longrightarrow & T^1.
\end{array}
$$

Now simply put $C^0 := C^2 + \{1\}$ (the original root). It remains to see that the map $t : T^1 \to T^0$ restricts to $C^1 \to C^0$, but this follows from the fact that if $t(b)$ is not in D^0, then it must be in either C^2 or 1. Using simple set theory, one readily checks that this is a tree again.

In any poset, we say that two elements e and e' are *incomparable* if neither $e \leq e'$ nor $e' \leq e$. If two subtrees have incomparable roots then they are disjoint. Indeed, suppose the subtrees S and S' of T have an edge x in common. Then the totally ordered set of ancestors of x in T will contain both the root of S and the root of S', hence they are comparable. Clearly siblings are incomparable. In particular, if two subtrees have sibling roots, then they are disjoint.

1.1.12 Lemma. *Let x and y be edges of a tree* T. *Then the following are equivalent:*

1. *The ideal subtrees D_x and D_y are disjoint.*

2. *x and y are incomparable (i.e. neither $x \leq y$ nor $y \leq x$).*

3. *There exists a subtree in which x and y are leaves.*

Proof. If $x \leq y$ then clearly $\mathsf{D}_x \subset \mathsf{D}_y$. On the other hand if D_x and D_y have an edge e in common, then $e \leq x$ and $e \leq y$, and hence $x \leq y$ or $y \leq x$. Concerning condition (3): if x and y are leaves of a subtree, in particular they are both minimal, and in particular they are incomparable. Conversely, if they are incomparable, then we already know that the ideal subtrees they generate are disjoint, so we can prune at x and y to get a subtree in which x and y are leaves. □

1.1.13. **Root-preserving embeddings.** An arrow $\mathsf{S} \to \mathsf{T}$ in ***TEmb*** is called *root preserving* if the root is mapped to the root. In other words, S viewed as a subtree of T contains the root edge of T:
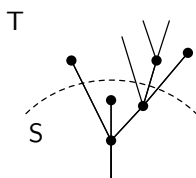
The root preserving subtrees are those that are up-sets in the tree order. It is easy to check that $\mathsf{S} \to \mathsf{T}$ is root-preserving if and only if the left-hand square is a pullback.

1.1.14 Lemma. *If a tree embedding is both root preserving and ideal, then it is invertible (i.e. its image is the whole tree).*

Proof. Indeed, if it is root preserving then its image contains 1, and because it is ideal its image contains all other edges, as they are descendants of the root. □

1.1.15 Proposition. *Every arrow $\phi : \mathsf{S} \to \mathsf{T}$ in **TEmb** factors uniquely as a root-preserving map followed by an ideal embedding.*

Proof. Put $r := \phi^0(\text{root})$, and consider the ideal subtree $\mathsf{D}_r \subset \mathsf{T}$. Since the map preserves the childhood relation, it is clear that all edges in S map into D_r, and this map is root preserving by construction. □

1.1.16. **Remark.** One can equally well factor every map the other way around: first an ideal embedding and then a root-preserving embedding. We will not have any use of that factorisation, though.

1.1.17 Lemma. *A subtree is determined by its boundary.*

Proof. Let $S \subset T$ and $S' \subset T$ be subtrees with common boundary. Suppose b is a node of S which is not in S'. Since b is in S, for some k we have $\sigma^k(t(b)) = \text{root}(S) = \text{root}(S')$. In this chain of nodes and edges, there is a node b which is in S but not in S', and such that $t(b)$ is an edge in S'. This means $t(b)$ is a leaf in S' and hence a leaf in S, but this in turn implies that b is not in S, in contradiction with the initial assumption. So the two subtrees contain the same nodes. If they do contain nodes at all then they are equal by Lemma 1.1.8. If both subtrees are trivial, then they must coincide because their roots coincide. $\square$

1.1.18. **Pushouts in *PolyEnd*.** A polynomial functor P is a diagram in *Set* of shape

$$\cdot \leftarrow \cdot \rightarrow \cdot \rightarrow \cdot$$

While pointwise sums are also sums in *PolyEnd*, pointwise pushouts are not in general pushouts in *PolyEnd*, because of the condition on arrows that the middle square be cartesian. Only pushouts over polynomial functors of shape ? $\leftarrow 0 \rightarrow 0 \rightarrow$? can be computed pointwise. In particular we can take pushouts over the trivial tree $\iota : 1 \leftarrow 0 \rightarrow 0 \rightarrow 1$. The pushout of the morphisms $S \leftarrow \iota \rightarrow T$ is the polynomial endofunctor given by

$$
\begin{array}{ccc}
S^2 + T^2 & \longrightarrow & S^1 + T^1 \\
\swarrow & & \searrow \\
S^0 + T^0 & & S^0 + T^0 \\
\swarrow & & \searrow \\
S^0 +_1 T^0 & & S^0 +_1 T^0,
\end{array}
\qquad (7)
$$

where $S^0 +_1 T^0$ denotes the amalgamated sum over the singleton.

1.1.19 Proposition. *Given a diagram of trees and tree embeddings*

$$ S \xleftarrow{\quad r \quad} \iota \xrightarrow{\quad l \quad} T $$

*such that r is the root edge in S, and l is a leaf in T, the pushout in **PolyEnd** is again a tree, called the grafting of S onto the leaf of T, and denoted $S +_\iota T$.*

Proof. We check that the polynomial endofunctor (7) is a tree by inspection of the four axioms. Axiom 1: it is obvious the involved sets are finite. Axiom 2: we check that the right-hand leg is injective: to say that l is a leaf of T means it is not in the image of $t : T^1 \rightarrow T^0$. So we can write $S^1 + T^1 = S^1 +_{\{l\}} (\{l\} + T^1)$, and the map we want to prove injective is just the inclusion $S^1 + T^1 = S^1 +_{\{l\}} (\{l\} + T^1) \hookrightarrow S^1 +_{\{l\}} T^0$. Axiom 3: we check that the left-hand leg is injective and has singleton complement: this follows from the calculation $S^0 +_1 T^0 = (S^2 + \{r\}) +_{\{r\}} T^0 = S^2 + T^0 = S^2 + T^2 + 1$ (where 1 denotes the root of the bottom tree T) Axiom 4: we check the walk-to-the-root condition: for $x \in S^0$, in a finite number of steps we arrive at $r = e = l$, and from here in another finite number of steps we come down to the root of T. $\square$

1.1.20 Remark. More generally, the pushout of a root-preserving embedding along an ideal embedding is again a tree, and the two resulting maps are again root-preserving and ideal, respectively, as in this diagram

We will not need or prove this result here.

The following expresses the recursive characterisation of trees.

1.1.21 Proposition. *A tree* T *is either a nodeless tree, or it has a node $b \in T^1$ with $t(b) = 1$; in this case for each $e \in (T^2)_b$ consider the ideal subtree D_e corresponding to e. Then the original tree* T *is the grafting of all the D_e onto the input edges of b.*

Proof. The grafting exists by Proposition 1.1.19, and is a subtree in T by the universal property of the pushout. Clearly every node in T is either *b* or a node in one of the ideal subtrees, therefore the grafting is the whole tree, by Lemma 1.1.8.   □

1.1.22 Corollary. *An automorphism of a tree amounts to permutation of siblings whose generated ideal subtrees are isomorphic.*

Proof. Use the recursive characterisation of trees. By childhood preservation, an automorphism must send an edge *e* to a sibling e'. For the same reason it must map D_e isomorphically onto $D_{e'}$.   □

1.1.23. **Inner edges.** An *inner edge* of a tree

$$T^0 \xleftarrow{\ s\ } T^2 \xrightarrow{\ p\ } T^1 \xrightarrow{\ t\ } T^0$$

is one that is simultaneously in the image of *s* and *t*. In other words, the set of inner edges is naturally identified with $T^1 \times_{T^0} T^2$ considered as a subset of T^0; its elements are pairs (b, e) such that $t(b) = s(e)$.

1.1.24 Corollary. *Every nontrivial tree* T *is the grafting (indexed by the set of inner edges $T^1 \times_{T^0} T^2$) of its one-node subtrees.*   □

The *elements* of a tree T are its nodes and edges. i.e. its elementary subtrees. These form a poset ordered by inclusion, and we denote this category el(T). There is an obvious functor el(T) → ***TEmb***. This functor has a colimit which is just T. Indeed, each edge is included in at most two one-node subtrees of T, and always as root in one and as leaf in the other; the colimit is obtained from these pushouts. The general notion of elements of a polynomial endofunctor will be introduced in Section 2.1.

1.2   P-**trees and free monads**

The trees studied so far are in a precise sense abstract trees, whereas many trees found in the literature are structured trees, amounting to a morphism to a fixed polynomial functor. The structure most commonly found is planarity: a planar structure on a tree T is a linear ordering of the input edges of each node, i.e. a linear ordering on each fibre of $T^2 \to T^1$. This amounts to giving a morphism $T \to M$, where M is the free-monoid monad (0.1.8).

1.2.1. P-**trees.** Let P be a fixed polynomial endofunctor $P^0 \leftarrow P^2 \to P^1 \to P^0$. By definition, the category of P-*trees* is the comma category ***TEmb*** $/P$ whose objects are trees T with a morphism $T \to P$ in ***PolyEnd***. Explicitly, a P-tree is a tree $T^0 \leftarrow T^2 \to T^1 \to T^0$ together with a diagram

$$
\begin{array}{ccccccc}
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0 \\
\downarrow & & \downarrow & \lrcorner & \downarrow & & \downarrow \\
P^0 & \longleftarrow & P^2 & \longrightarrow & P^1 & \longrightarrow & P^0 .
\end{array}
$$

Unfolding further the definition, we see that a P-tree is a tree whose edges are decorated in P^0, whose nodes are decorated in P^1, and with the additional structure of a bijection for each node $n \in T^1$ (with decoration $b \in P^1$) between the set of input edges of n and the fibre $(P^2)_b$, subject to the compatibility condition that such an edge $e \in (P^2)_b$ has decoration $s(e)$, and the output edge of n has decoration $t(b)$. Note that the P^0-decoration of the edges is completely specified by the node decoration together with the compatibility requirement, except for the case of a nodeless tree. (The notion of P-tree for a polynomial endofunctor P is closely related to the notion of T_S-tree of Baez and Dolan [1, Proof of Thm. 14], but they neglect to decorate the edge in the nodeless tree.)

   If P is the identity monad, a P-tree is just a linear tree. If P is the free-monoid monad, a P-tree is precisely a planar tree, as mentioned. If P is the free-nonsymmetric-operad monad on ***Set***$/\mathbb{N}$, the P-trees are the 3-dimensional opetopes, and so on: opetopes in arbitrary dimension are P-trees for a suitable P, cf. [1], [14, §7.1], [13].

1.2.2 Remark. It is important to note that P-trees are something genuinely different from just trees, in the sense that ***TEmb*** is not equivalent to ***TEmb***$/P$ for any P. It is true of course that every tree admits a planar structure, i.e. a decoration by the free-monoid monad M (0.1.8): the possible diagrams

$$
\begin{array}{ccccccc}
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0 \\
\downarrow & & \downarrow & \lrcorner & \downarrow & & \downarrow \\
I & \longleftarrow & \mathbb{N}' & \longrightarrow & \mathbb{N} & \longrightarrow & 1
\end{array}
$$

have to send a node $b \in T^1$ to its arity n (the number of input edges), and then there are $n!$ different choices for mapping the fibre to the n-element set **n**, the fibre over n.

The crucial property of P-trees is that they are rigid:

1.2.3 Proposition. P-*trees have no nontrivial automorphisms.*

Proof. Every automorphism of a tree consists in permuting siblings. Now in a P-tree, the set of siblings (some set $(T^2)_b$) is in specified bijection with $(P^2)_{\phi^1(b)}$, so no permutations are possible.

$\square$

The basic results about trees, notably grafting (1.1.19) and the recursive characterisation (1.1.21), have obvious analogues for P-trees. We shall not repeat those results.

1.2.4. **The set of** P**-trees.** Let P be a polynomial endofunctor. Denote by $\mathrm{tr}(\mathsf{P})$ the set of isomorphism classes of P-trees, i.e. isomorphism classes of diagrams

$$
\begin{array}{ccccccc}
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0 \\
\downarrow & & \downarrow & \lrcorner & \downarrow & & \downarrow \\
P^0 & \longleftarrow & P^2 & \longrightarrow & P^1 & \longrightarrow & P^0
\end{array}
$$

where the first row is a tree. Note that $\mathrm{tr}(\mathsf{P})$ is naturally a set over P^0 by returning the decoration of the root edge.

1.2.5 Theorem. *If* P *is a polynomial endofunctor then* $\mathrm{tr}(\mathsf{P})$ *is a least fixpoint (i.e. initial Lambek algebra) for the endofunctor*

$$
\begin{aligned}
1 + \mathsf{P} : \boldsymbol{Set}/P^0 & \longrightarrow & \boldsymbol{Set}/P^0 \\
X & \longmapsto & P^0 + \mathsf{P}(X).
\end{aligned}
$$

Proof. The proof uses the recursive characterisation of P-trees analogous to 1.1.21. For short, put $W := \mathrm{tr}(\mathsf{P})$. We have

$$
\mathsf{P}(W) = \left\{ (b, f) \mid b \in P^1, \; \begin{array}{c} (P^2)_b \xrightarrow{\ f\ } W \\ \searrow \quad \nearrow \\ P^0 \end{array} \right\}
$$

This set is in natural bijection with the set of P-trees with a root node decorated by $b \in P^1$. Indeed, given $(b, f) \in \mathsf{P}(W)$, we first consider the unique one-node P-tree whose node is decorated by b. (This is well-defined: since $(P^2)_b$ is finite, the one-node tree is given as in 1.0.5, and the decorations are completely determined by the requirement that the node is decorated by b.) Now for each $e \in (P^2)_b$ we can graft the P-tree $f(e)$ onto the leaf e of that one-node P-tree as in 1.1.19. The result is a P-tree D with root node decorated by b. Conversely, given a P-tree D with root node decorated by b, define $f : (P^2)_b \to W$ by sending e to the ideal sub-P-tree D_e.

Now, W is the sum of two sets: the nodeless P-trees (these are in bijection with P^0) and the P-trees with a root node. Hence we have $(1 + \mathsf{P})(W) \xrightarrow{\sim} W$, saying that W is a fixpoint.

Finally, we must show that W is a *least* fixpoint. Suppose $V \subset W$ is also a fixpoint. Let $W_n \subset W$ denote the set of P-trees with at most n nodes. Clearly $W_0 \subset V$. But if $W_n \subset V$ then also $W_{n+1} \subset V$ because each tree with $n+1$ nodes arises from some (b, f) where b decorates the root node and $f : (P^2)_b \to W_n$. $\qquad\qquad\qquad\square$

1.2.6. **Historical remarks: wellfounded trees.** Theorem 1.2.5 has a long history: it is a classical observation (due to Lambek) that the elements of an initial algebra for an endofunctor P are tree-like structures, and that the branching profile of such trees depends on P. A very general version of the theorem is due to Moerdijk and Palmgren [16] providing categorical semantics for the notion of W-types (wellfounded trees) in Martin-Löf type theory. Briefly, under the Seely correspondence between (extensional) type theory and locally cartesian closed categories $\mathscr{E}$, the Sigma and Pi types correspond to dependent sums and products (as in (3)). The W type constructor associates to a given combination P of Sigma and Pi types a new inductive type W_P. Under the correspondence, P is a polynomial endofunctor on $\mathscr{E}$ (i.e. with P^0 terminal), and W_P is its initial algebra.

The new feature of Theorem 1.2.5 (and the treatment leading to it) is to have trees and endofunctors on a common footing. This makes everything more transparent. Such a common footing was not possible in [16] because they only considered polynomial endofunctors P with P^0 terminal. Trees cannot be captured by such, since it is essential to be able to distinguish the edges in a tree. The case of arbitrary polynomial functors was considered by Gambino and Hyland [5], corresponding to dependent type theory.

1.2.7. **Construction of free monads.** Let $\mathrm{tr}'(\mathsf{P})$ denote the set of (isomorphism classes of) P-trees with a marked input leaf, i.e. the set of diagrams

$$
\begin{array}{ccccccc}
1 & \longleftarrow & 0 & \longrightarrow & 0 & \longrightarrow & 1 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
T^0 & \longleftarrow & T^2 & \longrightarrow & T^1 & \longrightarrow & T^0 \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
P^0 & \longleftarrow & P^2 & \longrightarrow & P^1 & \longrightarrow & P^0
\end{array}
$$

modulo isomorphism. (The cartesianness of the upper right-hand square says the edge is a leaf.) The set $\mathrm{tr}'(\mathsf{P})$ is naturally an object of $\mathbf{Set}/P^0$, the structure map $\mathrm{tr}'(\mathsf{P}) \to P^0$ returning the decoration of the marked leaf. There is also the natural projection to $\mathrm{tr}(\mathsf{P})$ given by forgetting the mark. We get altogether a polynomial functor

$$P^0 \leftarrow \mathrm{tr}'(\mathsf{P}) \to \mathrm{tr}(\mathsf{P}) \to P^0$$

which we denote by $\overline{\mathsf{P}}$. Its value on a set $X \to P^0$ is the set of P-trees with leaves decorated in X. More precisely, for a P-tree S, denote by L_S the set of leaves of S, then

$$\overline{P}(X) = \left\{ (\mathsf{S}, f) \mid \mathsf{S} \in \mathrm{tr}(\mathsf{P}), \ \begin{array}{c} L_\mathsf{S} \xrightarrow{\ f\ } X \\ \searrow \quad \swarrow \\ P^0 \end{array} \right\}.$$

The polynomial functor $\overline{\mathsf{P}}$ is naturally a monad: the multiplication map $\overline{\mathsf{P}}\,\overline{\mathsf{P}}(X) \to \overline{\mathsf{P}}(X)$ sends a P-tree T with leaves decorated by other P-trees to the P-tree obtained by grafting the other P-trees onto the leaves of T. Note that the compatibility condition on the decorations states that the root edges of the decorating trees are precisely the leaves of the bottom tree, so the grafting makes sense. The unit for the multiplication is the map $P^0 \to \overline{\mathsf{P}}(P^0)$ sending an edge x to the trivial P-tree decorated by $x \in P^0$.

The construction $\mathsf{P} \mapsto \overline{\mathsf{P}}$ is clearly functorial. If $\alpha : \mathsf{Q} \to \mathsf{P}$ is a morphism of polynomial endofunctors, it is clear that $\overline{\alpha}^1 : \mathrm{tr}(\mathsf{Q}) \to \mathrm{tr}(\mathsf{P})$ sends trivial trees to trivial trees, and it is also clear it is compatible with grafting. Hence $\overline{\alpha}$ is a monad map.

1.2.8 Proposition. *Let* P *be a polynomial endofunctor. The monad* $\overline{\mathsf{P}}$ *given by*

$$P^0 \leftarrow \mathrm{tr}'(\mathsf{P}) \to \mathrm{tr}(\mathsf{P}) \to P^0$$

is the free monad on P.

Proof. Given $X \to P^0$, put $W_X := \overline{\mathsf{P}}(X)$, the set of P-trees with leaves decorated in X. In other words,

$$W_X = \overline{\mathsf{P}}(X) = \left\{ (\mathsf{S}, f) \mid \mathsf{S} \in \mathrm{tr}(\mathsf{P}), \begin{array}{c} L_\mathsf{S} \xrightarrow{\ f\ } X \\ \searrow \ \ \nearrow \\ P^0 \end{array} \right\},$$

where L_S denotes the set of leaves of a tree S. It follows from the argument of Lemma 1.2.5 that W_X is a least fixpoint for the endofunctor $X + \mathsf{P}$, i.e. an initial object in $(X + \mathsf{P})$-***alg*** $\simeq X{\downarrow}P$-***alg***. Via the inclusion $\mathsf{P} \subset X + \mathsf{P}$ it also becomes a P-algebra. The construction $X \mapsto W_X$ is clearly functorial and defines a functor

$$\begin{array}{rcl} F : \mathbf{Set}/P^0 & \longrightarrow & \mathsf{P}\text{-}\mathbf{alg} \\ X & \longmapsto & W_X. \end{array}$$

To say that W_X is initial in $(X+\mathsf{P})$-***alg*** $\simeq X{\downarrow}P$-***alg*** is equivalent to saying that F is left adjoint to the forgetful functor $U : \mathsf{P}$-***alg*** $\to$ $\mathbf{Set}/P^0$, and therefore (e.g. by Barr-Wells [2, Theorem 4, p.311]), the generated monad $X \mapsto W_X$ is the free monad on P. $\square$

1.2.9. **The free monad on a tree.** We are particularly interested in the case where the polynomial endofunctor is itself a tree T. In this case we write $\mathrm{sub}(\mathsf{T})$ instead of $\mathrm{tr}(\mathsf{T})$, as we know that all maps between trees are injective. We restate this special case for emphasis:

1.2.10 Corollary. *Let* T *be a tree. The monad* $\overline{\mathsf{T}}$ *given by*

$$T^0 \leftarrow \mathrm{sub}'(\mathsf{T}) \to \mathrm{sub}(\mathsf{T}) \to T^0$$

is the free monad on T.

1.3   The category *Tree*

1.3.1. **The category *Tree*.** The category *Tree* is defined as the full subcategory of
PolyMnd consisting of the free monads $\overline{\mathsf{T}}$, where T is a tree. This means taking
the objects from *TEmb* and the morphisms from *PolyMnd*. More precisely the cat-
egory *Tree* is given by the Gabriel factorisation (identity-on-objects/fully-faithful) of
TEmb $\rightarrow$ *PolyMnd*:

$$
\begin{array}{ccc}
\textbf{\textit{Tree}} & \xrightarrow{\ \text{f.f.}\ } & \textbf{\textit{PolyMnd}} \\
{\scriptstyle\text{i.o.}}\Big\uparrow & & {\scriptstyle\text{free}}\Big\uparrow \dashv \Big\downarrow {\scriptstyle\text{forgetful}} \\
\textbf{\textit{TEmb}} & \xrightarrow{\quad\quad} & \textbf{\textit{PolyEnd}}
\end{array}
\tag{8}
$$

The category *Tree* is equivalent to the category Ω introduced by Moerdijk and Weiss [17],
whose presheaves are called dendroidal sets. The category *Tree* is also described as the
Kleisli category of the free-forgetful adjunction restricted to *TEmb*. The arrows in the
category *Tree* are by definition monad maps $\overline{\mathsf{S}} \rightarrow \overline{\mathsf{T}}$. By adjunction these correspond
to maps of endofunctors $\mathsf{S} \rightarrow \overline{\mathsf{T}}$, and many properties of the category *Tree* can be
extracted in this viewpoint, without ever giving an explicit description of the monad
maps. However, remarkably, the following result holds:

1.3.2 Proposition. *All maps of endofunctors* $\overline{\mathsf{S}} \rightarrow \overline{\mathsf{T}}$ *are monad maps. In other words, the
forgetful functor* ***Tree*** $\rightarrow$ ***PolyEnd*** *is full.*

This means that the maps in *Tree* have this surprisingly easy description: they are just
commutative diagrams

$$
\begin{array}{ccccccc}
S^0 & \longleftarrow & \text{sub}'(\mathsf{S}) & \longrightarrow & \text{sub}(\mathsf{S}) & \longrightarrow & S^0 \\
\Big\downarrow & & \Big\downarrow & & \Big\downarrow & & \Big\downarrow \\
T^0 & \longleftarrow & \text{sub}'(\mathsf{T}) & \longrightarrow & \text{sub}(\mathsf{T}) & \longrightarrow & T^0.
\end{array}
\tag{9}
$$

Proof of the Proposition. Since the monad structure is defined in terms of unit trees and
grafting, the assertion follows from the following two lemmas which are of indepen-
dent interest.

□

1.3.3 Lemma. *Any map of polynomial endofunctors* $\overline{\mathsf{S}} \rightarrow \overline{\mathsf{T}}$ *maps trivial subtrees to trivial
subtrees.*

Proof. If z is the root edge of a trivial subtree in S, then that trivial tree must map to a
subtree of T with root $\phi(z)$, by commutativity of the right-hand square in (9). On the
other hand, z is also the unique leaf of that trivial tree, and by commutativity of the
left-hand square in (9), the unique leaf of the image tree must be $\phi(z)$. Hence the image
tree has the property that its root is equal to its unique leaf, hence it is trivial.   □

1.3.4 Proposition. *Every morphism $\phi : \overline{S} \to \overline{T}$ respects grafting. In other words, if a subtree $R \in \mathrm{sub}(S)$ is a grafting $R = A +_\iota B$ then the image subtree $\phi^1(R) \in \mathrm{sub}(T)$ is given by $\phi^1(R) = \phi^1(A) +_{\phi^1(\iota)} \phi^1(B)$.*

Proof. The root of A is ι which is also a certain leaf of B. Hence the root of the image tree $\phi^1(A)$ is equal to $\phi^1(\iota)$ which is also a leaf of $\phi^1(B)$. Hence the grafting exists in T. It has root $\phi^0(\mathrm{root}(B))$ as required, and set of leaves $\phi^0(\mathrm{leaves}(A) + \mathrm{leaves}(B) \smallsetminus \{\iota\})$. So it has the same boundary as the image of R, so by Lemma 1.1.17 they agree. $\square$

1.3.5 Lemma. *A map of polynomial endofunctors $\overline{S} \to \overline{T}$ is completely determined by its value on the edge set.*

Proof. Let $R \subset S$ be an element of $\mathrm{sub}(S)$. The root of $\phi^1(R)$ must be the image of the root of R, by commutativity of the right-hand square of the representing diagram. Similarly, the set L_R of leaves of R is in bijection with the set of leaves of the image tree $\phi^1(R)$, by the cartesian condition on the middle square, but the latter set is also the image set $\phi(L_R)$, by commutativity of the left-hand square. Hence the set of leaves of $\phi^1(R)$ are fixed, so altogether the boundary of $\phi^1(R)$ is completely determined, and we conclude by Lemma 1.1.17. $\square$

1.3.6 Corollary. *If S is nontrivial, every map $\overline{S} \to \overline{T}$ is determined by its value on one-node subtrees. More precisely, the map is the grafting of maps on those one-node trees, indexed by the inner edges of S.*

Proof. The first statement follows because the images of the nodes determine the images of their output and input edges, hence all edges have their image determined by the images of the nodes. For the more precise statement, note that the tree S is the grafting of its one-node trees indexed by its inner edges (cf. 1.1.24). The inner edges map to edges again, and since grafting is preserved, the whole map $\phi : \overline{S} \to \overline{T}$ is the grafting of the restrictions of ϕ to the one-node subtrees (indexed by the inner edges of S). $\square$

1.3.7 Proposition. *Let ϕ be an arrow in **Tree**. Then ϕ^0 preserves the tree order:*

$$x \leq y \quad \Rightarrow \quad \phi^0(x) \leq \phi^0(y).$$

Furthermore, if x and y are incomparable, then $\phi^0(x)$ and $\phi^0(y)$ are incomparable.

Proof. Suppose $x \leq y$. Let S denote the minimal subtree with y as root edge and x as a leaf. Having x as marked leaf makes S an element in $\mathrm{sub}'(T)$. By construction, $s(S) = x$ and $t(p(S)) = y$. Now apply ϕ and use the fact that ϕ commutes with each of the structure maps. Hence $\phi^1(S)$ has $\phi^0(y)$ as root and $\phi^0(x)$ as marked leaf, and in particular $\phi^0(x) \leq \phi^0(y)$. For the second assertion, if x and y are incomparable, then by Lemma 1.1.12 there is a subtree in which x and y are leaves. Then $\phi^0(x)$ and $\phi^0(y)$ are leaves of the image subtree, and in particular incomparable. $\square$

Note that ϕ^0 is not distance preserving, though, and that it is not necessarily injective. When it is injective it also reflects the tree order.

1.3.8 Lemma. *If $\phi : \overline{S} \to \overline{T}$ is a map of trees, then $\phi^1 : \mathrm{sub}(S) \to \mathrm{sub}(T)$ is inclusion preserving.*

Proof. The statement is that if $Q \subset R$ are elements in $\mathrm{sub}(S)$ then we have $\phi^1(Q) \subset \phi^1(R)$ in $\mathrm{sub}(T)$. One way to see this is to observe that Q is determined by a subset of the nodes in R, cf. 1.1.8, and R is obtained from Q by grafting those complementary one-node trees onto Q. By preservation of grafting (1.3.4), $\phi^1(R)$ is therefore obtained from $\phi^1(Q)$ by grafting certain subtrees onto it, and in particular $\phi^1(Q) \subset \phi^1(R)$. □

We have now gathered some basic knowledge of what general maps in **Tree** look like, and we already had a firm grip on the maps in **TEmb**. The following proposition summarises various characterisations of the maps in **TEmb** from the viewpoint of **Tree**, that is, it characterises the free maps:

1.3.9 Proposition. *The following are equivalent for a map $\phi : \overline{S} \to \overline{T}$.*

1. *ϕ is free (i.e. of the form $\overline{\alpha} : \overline{S} \to \overline{T}$).*

2. *ϕ^0 is distance preserving.*

3. *The image of a one-node subtree is a one-node subtree.*

4. *For every subtree $R \subset S$, the image subtree $\phi^1(R) \subset T$ is isomorphic to R.*

5. *ϕ is injective, and if $R \in \mathrm{sub}(T)$ is hit by ϕ^1 then all the subtrees of R are hit too.*

6. *ϕ is injective, and if $R \in \mathrm{sub}(T)$ is hit by ϕ^1 then all edges of R are hit by ϕ^0.*

7. *ϕ is injective, and all edges in $\phi^1(S)$ are hit by ϕ^0.*

Proof. Straightforward verifications — omitted. □

1.3.10 Corollary. *In **Tree**, every isomorphism is free.* □

Another way to formulate Lemma 1.3.8 is that a map $\overline{S} \to \overline{T}$ restricts to any subtree $R \subset S$ to give a map $\overline{R} \to \overline{\phi(R)}$. This is in fact a key observation, featured in the next proposition.

1.3.11. **Boundary preserving maps.** A map $\phi : \overline{S} \to \overline{T}$ is called *boundary preserving* if it takes the maximal subtree to the maximal subtree. Equivalently, it takes leaves to leaves (bijectively) and root to root. It is clear that the composite of two boundary-preserving maps is boundary preserving, and that every isomorphism is boundary preserving.

1.3.12 Lemma. *Every surjection in **Tree** is boundary preserving.*

Proof. If $\phi^1 : \mathrm{sub}(S) \to \mathrm{sub}(T)$ is surjective, in particular the maximal subtree $T \in \mathrm{sub}(T)$ is hit, and since ϕ^1 is inclusion preserving by 1.3.8, $T \in \mathrm{sub}(T)$ must be hit by $S \in \mathrm{sub}(S)$. □

1.3.13 Proposition. *Every map of trees $\phi : \overline{S} \to \overline{T}$ factors essentially uniquely (i.e. uniquely up to unique isomorphism) as a boundary-preserving map followed by a free map. More precisely, the classes of boundary-preserving maps and free maps form an orthogonal factorisation system.*

We shall see in 2.2.9 that the boundary-preserving maps are precisely the *generic maps* in the sense of Weber [21] (defined in 2.2.1 below). Generic maps are characterised by a universal property. The proposition states that **Tree** has *generic factorisations*, an important property for a Kleisli category.

Proof. The first statement will be a special case of Proposition 2.2.7, but here is the main argument: let $M := \phi^1(S) \in \mathrm{sub}(T)$ denote the image of the maximal subtree in S, and let $\alpha : M \to T$ be the inclusion — this is a map in **TEmb**. Now $\overline{\alpha} : \overline{M} \to \overline{T}$ is the second factor in the wanted factorisation. Since ϕ^1 is inclusion preserving by Lemma 1.3.8, we get also a map $\overline{S} \to \overline{M}$ which is boundary preserving by construction. It is easy to see that this factorisation is unique (up to a unique isomorphism). Finally, since both classes of maps contain the isomorphisms and are closed under composition, we have an orthogonal factorisation system. □

1.3.14 Remark. There is a strong analogy between this boundary-preserving/free factorisation system in **Tree** and the root-preserving/ideal factorisation system in **TEmb**: in both cases the left-hand component is characterised in terms of a certain max-preservation, while the right-hand component is characterised in terms of stability with respect to smaller elements, or equivalently in terms of preservation of certain minimal elements. Compare Lemma 1.1.10 with Proposition 1.3.9. We shall not pursue the analogy further.

We shall now describe the boundary-preserving maps in explicit terms. The main point of the analysis is Proposition 1.3.4 which says that any map out of a nontrivial tree is the grafting of its restriction to the one-node subtrees (also via Corollary 1.1.24). This is also just the content of the adjunction: to give a map $\overline{S} \to \overline{T}$ is equivalent to giving $S \to \overline{T}$, so we just have to say where each node goes.
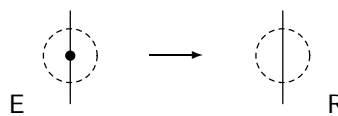
1.3.15. **Boundary-preserving maps out of a one-node tree.** Let E denote a one-node tree with n leaves, and suppose $\phi : \overline{E} \to \overline{R}$ is boundary preserving. By the cartesian condition, R necessarily has n leaves, and for any such R there are precisely $n!$ boundary-preserving maps from $\overline{E}$ to $\overline{R}$.

There are three cases to consider, depending on the number of nodes in R: If R has at least two nodes, then it has an inner edge, and since the map is boundary preserving this inner edge is not hit by ϕ^0, so ϕ is not surjective. Since in R the root is different from any leaf, ϕ^0 is injective, and therefore ϕ is injective. Here is a picture of such a *node refinement*:
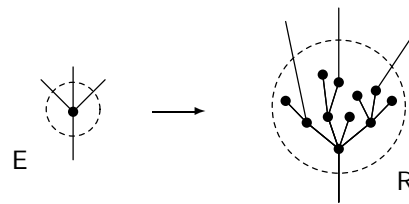
If R has precisely one node, clearly the map is an isomorphism. (This is not worth a picture.)

Finally there is a special case which occurs only for $n = 1$: then the tree R may be the trivial tree. In this case the two edges of E are both sent to the unique edge of R, and the node is sent to the maximal subtree (also trivial) by the boundary-preservation assumption. In this case, ϕ is clearly surjective (and not injective). Here is a picture of such a *unary-node deletion*:

1.3.16. **Boundary-preserving maps, general case.** Consider now a general boundary-preserving map $\overline{S} \to \overline{T}$, and assume S is nontrivial. We know the map is the grafting of its restrictions to the one-node subtrees of S. Let E be a one-node subtree of S. We can factor the composite $E \to \overline{S} \to \overline{T}$ into boundary-preserving followed by free:

$$
\begin{array}{ccc}
\overline{E} & \longrightarrow & \overline{S} \\
\text{b.pres.}\downarrow & & \downarrow \\
\overline{R} & \underset{\text{free}}{\longrightarrow} & \overline{T}.
\end{array}
$$

The subtree $R \in \mathrm{sub}(T)$ is the image of the subtree $E \in \mathrm{sub}(S)$. The map $\overline{E} \to \overline{R}$ is either a node refinement or a unary-node deletion or an isomorphism. The original map $\overline{S} \to \overline{T}$ is the grafting of all the maps $\overline{E} \to \overline{R}$ for E running over the set of nodes in S. Here is a picture:
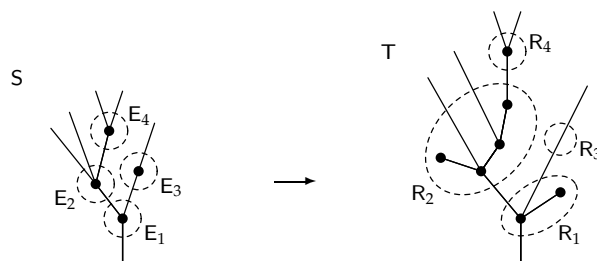
In conclusion, every boundary-preserving map $\overline{S} \to \overline{T}$ is the grafting of node refinements, unary-node deletions, and isomorphisms, indexed over the set of nodes in S. It is clear that we can realise the grafting by refining (or deleting) the nodes one by

one in any order, and in particular we can first do all the unary-node deletions (this amounts to a surjection), then all the node refinements (this amounts to an injection).

Since surjections are boundary-preserving (1.3.12), and since node refinements are not surjective we find:

1.3.17 Lemma. *Every map in **Tree** factors essentially uniquely as a surjection followed by an injection. The surjections are generated by the unary-node deletions.*   □

Combining the two factorisation systems we get a double factorisation system:

1.3.18 Proposition. *Every morphism in **Tree** factors essentially uniquely as a surjection (a sequence of node deletions), followed by a boundary-preserving injection (a sequence of node refinements), followed by a free map (essentially a subtree inclusion).*   □

1.3.19. **Description of boundary-preserving injections into a given tree.** To finish this first part of the paper, we show how to break the boundary-preserving injections into primitive maps. We already observed that we can refine one node at a time, but these are not the primitive maps. In order to characterise the primitive boundary-preserving injections, we change the viewpoint, looking at maps into a given tree instead of out of it:

Fix a tree T, and suppose it has an inner edge $x = t(b) = s(e)$. We construct a new tree T/x by contracting the edge x, and exhibit a canonical boundary-preserving injection $\phi_x : T/x \to \overline{T}$:

$$
\begin{array}{ccccccc}
T^0 \smallsetminus \{x\} & \longleftarrow & T^2 \smallsetminus \{x\} & \longrightarrow & T^1/(b = p(e)) & \longrightarrow & T^0 \smallsetminus \{x\} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
T^0 & \longleftarrow & \mathrm{sub}'(T) & \longrightarrow & \mathrm{sub}(T) & \longrightarrow & T^0
\end{array}
\tag{10}
$$

The maps are all obvious, except possibly $T^1/(b = p(e)) \to \mathrm{sub}(T)$: this map sends the node $b = p(e)$ to the two-node subtree containing b and $p(e)$, and sends all other nodes to their corresponding one-node subtree. It is clear that ϕ_X is boundary preserving and injective. The tree T/x has one inner edge less than T. We can now repeat the process for any subset of the inner edges in T, and for each subset we get a different boundary-preserving injection into T.

Conversely, every boundary-preserving injection $S \to T$ arises in this way. Indeed, we already know that these boundary-preserving injections are glued together from node refinements. The inner edges of the image trees form precisely the subset of edges we need to contract in order to recover the tree S.

In conclusion, we have derived explicit descriptions for each of the four classes of maps. The surjections can be described more explicitly as deletion of unary nodes, and each surjection can be broken into a composite of maps deleting just one node. The boundary-preserving injections are described as node refinements, and each boundary-preserving injection can be broken into a sequence of 'primitive' refinements adding

just one new node. The free maps are the 'arity-preserving' tree embeddings, which also can be given add-one-node wise. The new node is added either at a leaf (in which case the map is root preserving), or at the root (in which case the map is an ideal embedding).

1.3.20. **Linear trees.** A *linear tree* is one in which every node has precisely one input edge. The full subcategory of **Tree** consisting of the linear trees is equivalent to the simplex category Δ. The factorisation systems restrict to Δ, recovering the well-known fact that every map in Δ factors uniquely as a surjection followed by a top-and-bottom-preserving injection, followed by distance-preserving injection. The primitive maps correspond to degeneracy and face maps in Δ, which motivates the terminology employed by Moerdijk and Weiss [17]. They call the unary-node deletions *degeneracy maps*. The primitive node refinements they call *inner face maps*, and the primitive tree embeddings *outer face maps*. The inner face maps play a crucial role in their theory, to express horn-filling conditions for dendroidal sets [18].

2   Polynomial endofunctors in terms of trees

Since we are now going to consider presheaves, for size reasons we choose a skeleton for each of the categories **TEmb** $\subset$ **Tree**, and denote them with a lowercase initial:

$$\textbf{tEmb} \subset \textbf{tree}.$$

Clearly these are small categories.

The embedding $i_0 : \textbf{tEmb} \to \textbf{PolyEnd}$ induces a nerve functor

$$
\begin{aligned}
N_0 : \textbf{PolyEnd} &\longrightarrow \textbf{PrSh}(\textbf{tEmb}) \\
\mathsf{P} &\longmapsto \mathrm{Hom}_{\textbf{PolyEnd}}(i_0(-), \mathsf{P}).
\end{aligned}
$$

Similarly, $i : \textbf{tree} \to \textbf{PolyMnd}$ induces a nerve functor

$$
\begin{aligned}
N : \textbf{PolyMnd} &\longrightarrow \textbf{PrSh}(\textbf{tree}) \\
\mathsf{P} &\longmapsto \mathrm{Hom}_{\textbf{PolyMnd}}(i(-), \mathsf{P})
\end{aligned}
$$

The goal of this second part is to characterise the image of these nerve functors.

2.0   Background on nerve theorems

We shall first recall the classical nerve theorem for categories, then review Weber's general framework for nerve theorems.

2.0.1. **The nerve theorem for categories, after Berger [3].** It is classical that a simplicial set $X : \Delta^{\mathrm{op}} \to \textbf{Set}$ is (isomorphic to) the nerve of a small category if and only if the 'Segal condition' holds: for each $n \geq 1$, the natural map $X_n \to X_1 \times_{X_0} \cdots \times_{X_0} X_1$ is

an isomorphism. In the viewpoint of Berger and Weber, the starting point is the free-category monad on $\boldsymbol{Grph} = \boldsymbol{PrSh}(0 \rightrightarrows 1)$, the category of directed (non-reflexive) graphs. The free category on a graph A has as objects the vertices of A and as arrows the finite paths in A, i.e. graph maps from the finite, nonempty, linear graphs $[n] = \{0 \to 1 \to \cdots \to n\}$ into A. Let $\Delta_0 \subset \boldsymbol{Grph}$ denote the full subcategory consisting of those linear graphs; note that all maps in Δ_0 are injective. Equipped with the jointly-surjective topology, there is an equivalence of categories

$$\boldsymbol{Sh}(\Delta_0) \simeq \boldsymbol{Grph} \tag{11}$$

(the topology is in fact determined by this equivalence).

Now Δ appears as the Kleisli category over Δ_0 with respect to the free-category monad, and we have the diagram

as in (8). The category Δ has generic-free factorisation; the generic maps are the endpoint-preserving maps. (The free maps can be characterised as distance preserving.) The embeddings i_0 and i are dense, hence induce fully faithful nerve functors

The classical nerve theorem can now be broken into two steps. The first says that $X : \Delta^{\mathrm{op}} \to \boldsymbol{Set}$ is in the essential image of N if and only if j^*X is in the essential image of N_0. The second step concerns presheaves on Δ_0, and can be phrased in several equivalent ways: $X : \Delta_0^{\mathrm{op}} \to \boldsymbol{Set}$ is a graph if and only if it is a sheaf (this is essentially a reformulation of the equivalence (11)). The sheaf condition amounts to the condition that certain cocones are sent to limit cones: each $[n] \in \Delta_0$ has a canonical expression as a colimit of copies of the two representables $[0]$ and $[1]$, and the condition amounts to saying that these cocones are sent to limit cones. This is just the usual Segal condition.

Berger [3] explained the nerve theorem for categories along these lines as a baby case of a similar theorem characterising strict ω-categories as presheaves-with-Segal-condition on the cell category Θ of Joyal [9]. Leinster [15] proved a more general nerve theorem (cf. 2.0.3), and Weber [21] fitted everything into a natural (and more general) framework whose main notions we now recall.

2.0.2. **Local right adjoints and generic factorisation.** (Cf. Weber [20].) A functor $F : \mathscr{D} \to \mathscr{C}$ is called a *local right adjoint* if every slice

$$
\begin{array}{ccc}
\mathscr{D}/X & \longrightarrow & \mathscr{C}/FX \\
[Y \to X] & \longmapsto & [FY \to FX]
\end{array}
$$

is a right adjoint. (If $\mathscr{D}$ has a terminal object, the notion of local right adjoint coincides with Street's notion of parametric right adjoint [19].) This condition is equivalent to the following: every map in $\mathscr{C}$ of the form $a : A \to FX$ factors essentially uniquely as

$$
\begin{array}{ccc}
A & \xrightarrow{\ a\ } & FX \\
 & \underset{g}{\searrow} \quad \nearrow_{Ff} & \\
 & FM &
\end{array}
$$

where $g : A \to FM$ is generic. We shall define generic in a moment (2.2.1).

Suppose now that $\mathscr{C} = \textbf{\textit{PrSh}}(\mathbb{C})$ and that F is a local right adjoint cartesian monad on $\mathscr{C}$. In this case the local adjointness condition is equivalent to having generic factorisations just for maps of the form $C \to F1$, where $C \in \mathbb{C}$ is a representable object in $\mathscr{C}$. Choose one such factorisation for each representable. Let Θ_0 denote the full subcategory of $\mathscr{C}$ consisting of the objects M appearing in these factorisations. Let Θ denote the Kleisli category of F restricted to Θ_0, i.e. the full subcategory of F-**Alg** given by the FM for $M \in \Theta_0$. We have a diagram

$$
\begin{array}{ccc}
\Theta & \xrightarrow{\ i\ } & F\textbf{-Alg} \\
j \downarrow & \quad \text{free} \dashv \text{forgetful} & \uparrow \\
\Theta_0 & \xrightarrow[i_0]{} & \mathscr{C}.
\end{array}
\tag{12}
$$

2.0.3. **Special nerve theorem.** (Cf. Leinster [15] and Weber [21].) *The nerve functor $N : F\textbf{-Alg} \to \textbf{\textit{PrSh}}(\Theta)$ induced by i is fully faithful, and its essential image consists of those presheaves on Θ satisfying the* Segal condition, *namely that the canonical cocones are sent to limit cones in* ***Set***.

2.0.4. **Monads with arities.** The conditions needed in order to get a nerve theorem have been further abstracted by Weber [21]. A *monad with arities* on a category $\mathscr{C}$ (not required to be a presheaf category or even to have a terminal object) consists of a monad F (not required to be cartesian) and a full subcategory $i_0 : \Theta_0 \subset \mathscr{C}$ required to be dense and small, such that the following condition is satisfied: the left Kan extension

$$
\begin{array}{ccc}
\Theta_0 & \xrightarrow{\ i_0\ } & \mathscr{C} \\
i_0 \searrow & \overset{\text{id}}{\Rightarrow} \quad \nearrow_{\text{id}} & \\
 & \mathscr{C} &
\end{array}
$$

is preserved by the composite

$$\mathscr{C} \xrightarrow{\;T\;} \mathscr{C} \xrightarrow{\;N_0\;} \textbf{\textit{PrSh}}(\Theta_0)$$

(cf. the proof of Proposition 2.2.10 below). As above, denote by Θ the Kleisli category of F restricted to Θ_0, refer to Diagram (12), and let $N : F\textbf{\textit{-Alg}} \to \textbf{\textit{PrSh}}(\Theta)$ and $N_0 : \mathscr{C} \to \textbf{\textit{PrSh}}(\Theta_0)$ be the nerve functors induced by i and i_0 respectively.

2.0.5. **General nerve theorem.** (Cf. Weber [21, Thm. 4.10].) *If (F, Θ_0) is a monad with arities, then N is fully faithful, and $X : \Theta^{\mathrm{op}} \to \textbf{\textit{Set}}$ is in the essential image of N if and only if its restriction to Θ_0 is in the essential image of N_0.*

The remainder of this article is concerned with establishing a nerve theorem for polynomial endofunctors and polynomial monads. In this case, the Segal condition is not enough to characterise the nerves, the reason being that **PolyEnd** is not a presheaf category: specifically it lacks a terminal object.

2.1 Elements of a polynomial endofunctor

Although **PolyEnd** is not a presheaf category, we shall see in a moment that all its slices are presheaf categories. The crucial construction is that of a category of elements of a polynomial endofunctor.

Recall that for an object F of a presheaf category $\textbf{\textit{PrSh}}(\mathbb{C})$, the category of elements is the comma category $\mathbb{C}/F$, via the Yoneda embedding. Among the important properties of this construction is the fact that the composite functor

$$\mathbb{C}/F \longrightarrow \mathbb{C} \xrightarrow{\;y\;} \textbf{\textit{PrSh}}(\mathbb{C})$$

has colimit F; it is called the *canonical diagram* for F. Second, there is a canonical equivalence of categories
$$\textbf{\textit{PrSh}}(\mathbb{C})/F \simeq \textbf{\textit{PrSh}}(\mathbb{C}/F).$$

We shall introduce the category of elements of a polynomial endofunctor P, and establish the analogues of these two properties.

2.1.1. **The category of elements of a polynomial endofunctor.** To a given polynomial endofunctor P, represented by $I \leftarrow E \to B \to I$, we associate a small category el(P), the *category of elements* of P. Since **Poly** is not a presheaf category, this notion is not the standard notion of category of elements. However, the terminology will be justified by Proposition 2.1.4 below. We first associate a bipartite graph to P: its vertex set is $I + B$ and every edge will go from an element in I to an element in B. The set of edges is $B + E$. An edge $b \in B$ has source $t(b)$ and target b. An edge $e \in E$ has source $s(e)$ and target $p(e)$. Now define el(P) to be the category generated by that bipartite graph; since the graph is bipartite, there are no composable arrows, so this step just

amounts to adding an identity arrow for each object. It is clear that this construction is functorial, so we have defined a functor

$$\text{el} : \textbf{\textit{PolyEnd}} \to \textbf{\textit{Cat}},$$

easily seen to be faithful.

2.1.2. Examples. Let P be the identity functor (represented by $1 \leftarrow 1 \to 1 \to 1$). Then el(P) is a category with two objects and two parallel arrows (in addition to the identity arrows): $0 \rightrightarrows 1$. That is, the category whose presheaves are the graphs.

Let P be the free-monoid endofunctor M (represented by $1 \leftarrow \mathbb{N}' \to \mathbb{N} \to 1$, cf. 0.1.8). Then el(M) has object set $1 + \mathbb{N} = \{\iota, 0, 1, 2, \dots\}$, and its arrows all go from ι to somewhere else. The set of arrows is $\mathbb{N} + \mathbb{N}'$ (plus the identity arrows): for each $n \in \mathbb{N}$ there is an arrow $\iota \to n$, and for each $\{i < n\} \in \mathbb{N}'$ there is another arrow $\iota \to n$. This is the category whose presheaves are the *nonsymmetric coloured collections* (2.4.4) (called multigraphs in [21], Example 2.14).

2.1.3 Proposition. *There is an equivalence of categories*

$$\textbf{\textit{PolyEnd}}/P \simeq \textbf{\textit{PrSh}}(\text{el}(P)).$$

Proof. To a given polynomial endofunctor Q over P:

assign the presheaf $\widetilde{Q} : \text{el}(P)^{\text{op}} \to \textbf{\textit{Set}}$ whose value on the object i is A_i, whose value on the object b is N_b, whose value on the arrow $t(b) \to b$ is the restriction $N_b \to A_{t(b)}$, and whose value on the arrow $e : s(e) \to p(e)$ is the composite $N_{p(e)} \simeq M_e \to A_{s(e)}$.

Conversely, given a presheaf $X : \text{el}(P)^{\text{op}} \to \textbf{\textit{Set}}$, we have in particular sets $X(i)$ or $X(b)$ for each of the objects i or b of $I + B$. Define a polynomial functor over P by

For each $b \in B$ there is an arrow $b : t(b) \to b$ in el(P) and hence a map $X(b) \to X(t(b))$, the sum of these maps constitute the endpoint component of the map. For each $e \in E$ there is an arrow $e : s(e) \to p(e)$ in el(P) and hence a map $X(p(e)) \to X(s(e))$; the sum of these maps constitute the left-hand map of the polynomial functor. The map in the middle is obvious.

It is easy to see that these assignments are functorial, and that the two constructions are inverse to each other. □

2.1.4 Proposition. *Let* Q *be a polynomial endofunctor over* P, *and let* $\widetilde{Q} : \mathrm{el}(P)^{\mathrm{op}} \to$ ***Set*** *denote the corresponding presheaf via Proposition 2.1.3. Then there is a natural isomorphism between* $\mathrm{el}(Q)$ *and the category of elements of* $\widetilde{Q}$.

Proof. (We work just with the graphs, omitting the verification needed for identity arrows.) Let $Q \to P$ be given by

$$
\begin{array}{ccccccc}
A & \longleftarrow & M & \longrightarrow & N & \longrightarrow & A \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
I & \underset{s}{\longleftarrow} & E & \underset{p}{\longrightarrow} & B & \underset{t}{\longrightarrow} & I.
\end{array}
$$

The elements of the presheaf $\widetilde{Q} : \mathrm{el}(P)^{\mathrm{op}} \to$ ***Set*** are pairs (u, x) where $u \in \mathrm{obj}(\mathrm{el}(P)) = I + B$ and $x \in \widetilde{Q}(u)$. In other words, an element of $\widetilde{Q}$ is either a pair (i, a), with $i \in I$ and $a \in \widetilde{Q}(i) = A_i$, or a pair (b, n), with $b \in B$ and $n \in \widetilde{Q}(b) = N_b$. In conclusion the object set is $A + N$, as desired.

An arrow from element (u, x) to element (v, y) is a pair (f, y) where $f : u \to v$ belongs to $\mathrm{arr}(\mathrm{el}(P)) = B + E$, and $y \in \widetilde{Q}(v)$. In other words, an arrow is either a pair (b, n) where $b : t(b) \to b$ in $\mathrm{el}(P)$ and $n \in \widetilde{Q}(b) = N_b$, or it is a pair (e, m) where $e : s(e) \to p(e)$ in $\mathrm{el}(P)$ and $m \in \widetilde{Q}(p(e)) = N_{p(e)} \simeq M_e$. In conclusion, the set of arrows of the category of elements of $\widetilde{Q}$ is $N + M$, and it is clear from the construction that their sources and targets are as required. $\qquad\square$

2.1.5. Elementary trees. Let ***ElTr*** denote the full subcategory of ***TEmb*** consisting of the elementary trees, i.e. trees with at most one node. Let ***elTr*** denote a fixed skeleton of ***ElTr***: we fix one trivial tree ι and one one-node tree n for each $n \in \mathbb{N}$. There are $n + 1$ arrows from ι to n, one for each leaf and one for the root edge, and in addition to these arrows, each object n has $n!$ endomorphisms, all invertible. Henceforth we use the symbol n to denote an arbitrary object of ***elTr***, possibly the trivial tree.

For each polynomial endofunctor P, there is a canonical equivalence of categories

$$
\mathrm{el}(P) \simeq \textbf{\textit{ElTr}}/P : \tag{13}
$$

to each element $i \in I$ corresponds the trivial P-tree

$$
\begin{array}{ccccccc}
\{i\} & \longleftarrow & 0 & \longrightarrow & 0 & \longrightarrow & \{i\} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
I & \longleftarrow & E & \longrightarrow & B & \longrightarrow & I,
\end{array}
$$

and to each element $b \in B$ corresponds the one-node P-tree

$$
\begin{array}{ccccccc}
E_b + \{t(b)\} & \longleftarrow & E_b & \longrightarrow & \{b\} & \longrightarrow & E_b + \{t(b)\} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
I & \longleftarrow & E & \longrightarrow & B & \longrightarrow & I.
\end{array}
$$

In particular, each element of P can be viewed as a morphism n → P (in analogy with the situation for presheaves), and we can think of **ElTr** as the category of representables. (We shall come back to the relation between **PolyEnd** and the presheaf category **PrSh**(**elTr**) in Section 2.4.)

The equivalences we have established fit into this diagram (commutative up to isomorphism):

$$
\begin{array}{ccc}
\mathrm{el(P)} & \xrightarrow{\;y\;} & \textbf{\textit{PrSh}}\,(\mathrm{el(P)}) \\
\simeq \Big\downarrow & & \Big\downarrow \simeq \\
\textbf{\textit{elTr}}\,/\mathrm{P} & \longrightarrow & \textbf{\textit{PolyEnd}}\,/\mathrm{P},
\end{array}
$$

so locally **elTr** → **PolyEnd** *is* the Yoneda embedding of the representables.

2.1.6. **The canonical diagram for** P. The composite functor

$$
D_{\mathrm{P}} : \quad \mathrm{el(P)} \longrightarrow \textbf{\textit{PolyEnd}}\,/\mathrm{P} \longrightarrow \textbf{\textit{PolyEnd}}
$$

is called the *canonical diagram* for P, in view of the following result:

2.1.7 Proposition. *The colimit of D_{P} is* P.

Proof. For Q over P we have

$$
\mathrm{colim}\,\big(\,\mathrm{el(Q)} \to \textbf{\textit{PolyEnd}}\,/\mathrm{P}\,\big) \;=\; \mathrm{Q}
$$

since **PolyEnd**/P is a presheaf category by 2.1.3, and since the notion of elements is the usual one in this case by 2.1.4. The result now follows since the forgetful functor **PolyEnd**/P → **PolyEnd** preserves colimits. (It is a basic fact that **PolyEnd** has binary products, cf. [12], so P × − is right adjoint to the forgetful functor in question.)   □

2.2   Generic factorisation, and trees as arities

The main results of this section are that the free-monad monad on **PolyEnd** is a local right adjoint, or equivalently that it has generic factorisations, and that trees are arities for the free-monad monad. We first show that elements of free monads have generic factorisation and that the resulting middle objects are precisely the trees. This is an explicit verification relying on our good handle on trees and free monads. The second step is to establish local right adjointness from the first result using the density of elementary trees. Finally we find that for trees the notions of generic and boundary-preserving coincide, as claimed in Section 1.3. From this it follows readily that trees satisfy Weber's axioms for arities.

2.2.1. **Generic maps.** The notion of generic map was introduced by Weber [20] generalising the notion of generic element of [8]. The following special case of the notion is

the most useful. Let F be a monad on a category $\mathscr{C}$. An arrow $g : A \to FB$ in $\mathscr{C}$ is called *generic* (with respect to F) if for every diagram

$$
\begin{array}{ccc}
A & \longrightarrow & FC \\
g \downarrow & & \downarrow F(c) \\
FB & \xrightarrow{F(b)} & FD
\end{array}
$$

there is a unique arrow $u : B \to C$ such that

$$
\begin{array}{ccc}
 & C & \\
u \nearrow & \downarrow c & \\
B \xrightarrow{b} & D &
\end{array}
\qquad \text{and} \qquad
\begin{array}{ccc}
A & \longrightarrow & FC \\
g \downarrow & \nearrow F(u) & \\
FB & &
\end{array}
$$

The endofunctor F is said to *admit generic factorisations* if every $A \to FD$ admits a factorisation into a generic map followed by one in the image of F. (This condition is equivalent to being a local right adjoint.) Such a factorisation is then necessarily unique up to unique isomorphism. The maps of the form $A \to FD$ correspond by adjunction to F-algebra maps $FA \to FD$, which we shall also call generic. All the involved maps can then be seen as living in the Kleisli category of F. From this perspective, generic factorisations amount to having an orthogonal factorisation system generic/free in the Kleisli category. (To be correct, the right hand class of the factorisation system should be saturated with the isomorphisms. In our case all isomorphisms will already be free.)

2.2.2 Lemma. *If $g : \overline{\mathsf{S}} \to \overline{\mathsf{T}}$ is a boundary-preserving map between trees, then g is generic in* **PolyEnd** *(with respect to the free-monad monad $\mathsf{P} \mapsto \overline{\mathsf{P}}$).*

Proof. Given a square

$$
\begin{array}{ccc}
\overline{\mathsf{S}} & \xrightarrow{\sigma} & \overline{\mathsf{Q}} \\
g \downarrow & & \downarrow \overline{q} \\
\overline{\mathsf{T}} & \xrightarrow{\overline{\tau}} & \overline{\mathsf{P}}
\end{array}
$$

let $\mathsf{R} \to \mathsf{Q}$ denote the Q-tree $\sigma^1(\mathsf{S})$. It image under $\overline{q}^1$ is just $\mathsf{R} \to \mathsf{Q} \to \mathsf{P}$. Going the other way around the square, the maximal subtree S is first mapped to $\mathsf{T} \in \mathrm{sub}(\mathsf{T})$ (by boundary preservation), and then to the P-tree $\tau : \mathsf{T} \to \mathsf{P}$. Since the diagram commutes, R and T represent the same P-tree, and since P-trees have no nontrivial automorphisms, there is a unique isomorphism $\mathsf{T} \overset{\sim}{\to} \mathsf{R}$, and hence a unique diagonal filler $d : \mathsf{T} \to \mathsf{Q}$ for the square. $\qquad\qquad \square$

We shall see in a moment that the converse of the lemma is true as well.

2.2.3. **Generic factorisation of elements of free monads.** The key point towards getting all generic factorisations is the following easy result:

2.2.4 Lemma. *Every element $s : \mathsf{n} \to \overline{\mathsf{P}}$ of a free monad $\overline{\mathsf{P}}$ factors essentially uniquely as*

$$\overline{\mathsf{n}} \xrightarrow{\ g\ } \overline{\mathsf{T}} \xrightarrow{\ \overline{f}\ } \overline{\mathsf{P}},$$

where the middle object T is a tree, g is boundary preserving (hence generic), and $\overline{f}$ is free.

Proof. If n is the trivial tree, the map $s : \mathsf{n} \to \overline{\mathsf{P}}$ singles out a single element $x \in P^0$, and s is the free map

$$
\begin{array}{ccccccc}
1 & \longleftarrow & 0 & \longrightarrow & 0 & \longrightarrow & 1 \\
{\scriptstyle \ulcorner x \urcorner}\big\downarrow & & \big\downarrow\ \lrcorner & & \big\downarrow & & \big\downarrow {\scriptstyle \ulcorner x \urcorner} \\
P^0 & \longleftarrow & P^2 & \longrightarrow & P^1 & \longrightarrow & P^0.
\end{array}
$$

Hence we can take $\mathsf{T} = \mathsf{n}$, and g is just the identity.

If n is a one-node tree, the unique element in n^1 maps to some element in $(\overline{P})^1 = \mathrm{tr}(P)$, i.e. a P-tree $f : \mathsf{T} \to \mathsf{P}$. The cartesian condition on maps ensures there is a bijection between the input edges of the unique node in n and the leaves of T, hence a unique map $\overline{\mathsf{n}} \to \overline{\mathsf{T}}$ making the triangle commute — clearly this map is boundary preserving. $\qquad\square$

2.2.5. **The spine.** Fix a polynomial endofunctor P, and assume a choice of a generic factorisation for every element of $\overline{\mathsf{P}}$, as in Lemma 2.2.4. These factorisations fit together to define a functor

$$E_{\mathsf{P}} : \mathrm{el}(\overline{\mathsf{P}}) \to \boldsymbol{PolyEnd}/\mathsf{P}$$

(sometimes called the *spine* relative to P) sending an element of $\overline{\mathsf{P}}$ to the tree appearing in the factorisation. Up to isomorphism, every P-tree $\mathsf{T} \to \mathsf{P}$ arises like this: just precompose $\mathsf{T} \to \mathsf{P}$ with a boundary-preserving map from the one-node tree with the same number of leaves as T. (Note that if n is unary, its node may be mapped to a trivial P-tree. Hence the functor E_{P} is not injective on objects, not even on isomorphism classes of objects.)

2.2.6 Proposition. *The monad $\mathsf{P} \mapsto \overline{\mathsf{P}}$ is a local right adjoint. That is, for each P, the natural functor*

$$
\begin{array}{rcl}
\boldsymbol{PolyEnd}/\mathsf{P} & \longrightarrow & \boldsymbol{PolyEnd}/\overline{\mathsf{P}} \\
[\mathsf{Q} \to \mathsf{P}] & \longmapsto & [\overline{\mathsf{Q}} \to \overline{\mathsf{P}}]
\end{array}
\tag{14}
$$

has a left adjoint.

Proof. The asserted left adjoint will be the left Kan extension of E_P along the Yoneda embedding $y : \mathrm{el}(\overline{\mathsf{P}}) \to \boldsymbol{PrSh}(\mathrm{el}(\overline{\mathsf{P}})) \simeq \boldsymbol{PolyEnd}/\overline{\mathsf{P}}$:

$$
\begin{array}{ccc}
\mathrm{el}(\overline{\mathsf{P}}) & \xrightarrow{\;\;y\;\;} & \boldsymbol{PolyEnd}/\overline{\mathsf{P}} \\
& E_\mathsf{P} \searrow \qquad \swarrow \mathrm{lan}_y E_\mathsf{P} & \\
& \boldsymbol{PolyEnd}/\mathsf{P} &
\end{array}
$$

The functor $\mathrm{lan}_y E_\mathsf{P}$ sends a polynomial endofunctor $\mathsf{F} \to \overline{\mathsf{P}}$ to the colimit of the functor

$$
\mathrm{el}(\mathsf{F}) \longrightarrow \mathrm{el}(\overline{\mathsf{P}}) \xrightarrow{\;E_\mathsf{P}\;} \boldsymbol{PolyEnd}/\mathsf{P}.
$$

Using the identification $\boldsymbol{PolyEnd}/\overline{\mathsf{P}} \simeq \boldsymbol{PrSh}(\mathrm{el}(\overline{\mathsf{P}}))$, it is a general fact that $\mathrm{lan}_y(E_\mathsf{P})$ has a right adjoint

$$
\begin{array}{rcl}
\mathrm{res} : \boldsymbol{PolyEnd}/\mathsf{P} & \longrightarrow & \boldsymbol{PrSh}(\mathrm{el}(\overline{\mathsf{P}})) \\
[q : \mathsf{Q} \to \mathsf{P}] & \longmapsto & \left[[s : \mathsf{n} \to \overline{\mathsf{P}}] \mapsto \mathrm{Hom}_{\boldsymbol{PolyEnd}/\mathsf{P}}(E_\mathsf{P}(s), q)\right].
\end{array}
$$

So to establish the claim we must show that res is isomorphic to (14). Fix $q : \mathsf{Q} \to \mathsf{P}$. The monad $\overline{q} : \overline{\mathsf{Q}} \to \overline{\mathsf{P}}$ corresponds (under the equivalence of 2.1.3) to the presheaf

$$
\begin{array}{rcl}
\mathrm{el}(\overline{\mathsf{P}})^{\mathrm{op}} & \longrightarrow & \boldsymbol{Set} \\
[s : \mathsf{n} \to \overline{\mathsf{P}}] & \longmapsto & \mathrm{Hom}_{\boldsymbol{PolyEnd}/\overline{\mathsf{P}}}(s, \overline{q}).
\end{array}
$$

But the required bijection,

$$
\mathrm{Hom}_{\boldsymbol{PolyEnd}/\mathsf{P}}(E_\mathsf{P}(s), q) \simeq \mathrm{Hom}_{\boldsymbol{PolyEnd}/\overline{\mathsf{P}}}(s, \overline{q})
$$

is precisely the factorisation property established in Lemma 2.2.4: in a diagram

$$
\begin{array}{ccc}
\mathsf{n} & \dashrightarrow & \overline{\mathsf{Q}} \\
\text{generic} \downarrow & \nearrow & \downarrow \overline{q} \\
\overline{\mathsf{T}} & \xrightarrow[E_\mathsf{P}(s)]{} & \overline{\mathsf{P}},
\end{array}
$$

giving the top arrow $s \to \overline{q}$ (i.e. a map $\mathsf{n} \to \overline{\mathsf{Q}}$ over $\overline{\mathsf{P}}$) is equivalent to giving the diagonal filler $E_\mathsf{P}(s) \to q$ (i.e. a map $\mathsf{T} \to \mathsf{Q}$ over P). □

2.2.7 Proposition. *Let T be a tree and P an arbitrary polynomial endofunctor. Every monad map $h : \overline{\mathsf{T}} \to \overline{\mathsf{P}}$ factors essentially uniquely as*

$$
\begin{array}{ccc}
\overline{\mathsf{T}} & \xrightarrow{\;\;h\;\;} & \overline{\mathsf{P}} \\
g \searrow & & \nearrow \overline{f} \\
& \overline{\mathsf{M}} &
\end{array}
$$

where M is a tree, $g : \overline{\mathsf{T}} \to \overline{\mathsf{M}}$ is boundary preserving (hence generic), and $\overline{f} : \overline{\mathsf{M}} \to \overline{\mathsf{P}}$ is free.

Proof. Consider the map $h^1 : \mathrm{sub}(\mathsf{T}) \to \mathrm{tr}(\mathsf{P})$ and let $f : \mathsf{M} \to \mathsf{P}$ be the image of $\mathsf{T} \in \mathrm{sub}(\mathsf{T})$. If T is the trivial tree, h is already free, and $\mathsf{M} = \mathsf{T}$. Otherwise we construct $g : \overline{\mathsf{T}} \to \overline{\mathsf{M}}$ by grafting: each one-node subtree $\mathsf{S} \in \mathrm{sub}(\mathsf{T})$ is mapped by h^1 to some P-tree $\mathsf{R} := h^1(\mathsf{S})$, and there is a unique boundary-preserving tree map $\overline{\mathsf{S}} \to \overline{\mathsf{R}}$. The map g is the grafting of all these maps (indexed by the inner edges of T): since h as a monad map preserves grafting, the grafting of the P-trees R is precisely M. Uniqueness of the factorisation follows from Lemma 2.2.2. □

2.2.8 Corollary. *If T is a tree and $g : \overline{\mathsf{T}} \to \overline{\mathsf{R}}$ is generic in **PolyEnd** (with respect to the free-monad monad $\mathsf{P} \mapsto \overline{\mathsf{P}}$), then R is a tree and g is boundary preserving.*

Proof. Factor g as boundary preserving followed by free: $\overline{\mathsf{T}} \to \overline{\mathsf{M}} \to \overline{\mathsf{R}}$, where T is a tree. Since boundary-preserving maps between trees are generic, and by uniqueness of generic factorisations, we have $\mathsf{M} \simeq \mathsf{R}$, hence g is boundary preserving. □

2.2.9 Corollary. *In the category **Tree**, the generic maps are precisely the boundary-preserving maps.* □

2.2.10 Proposition. *The subcategory **tEmb** $\subset$ **PolyEnd** provides arities for the free-monad monad $F : $ **PolyEnd** $\to$ **PolyEnd**.*

Proof. We have already shown that the free-monad monad is a local right adjoint, and that the subcategory **tEmb** $\subset$ **PolyEnd** is (small and dense and) closed under generic factorisation. It remains to establish that the left Kan extension

$$
\begin{array}{ccc}
\textbf{tEmb} & \xrightarrow{\ i_0\ } & \textbf{PolyEnd} \\
& \underset{i_0}{\searrow} \quad \overset{\mathrm{id}}{\Rightarrow} \quad \nearrow {\scriptstyle\mathrm{id}} & \\
& \textbf{PolyEnd} &
\end{array}
$$

is preserved by the composite

$$
\textbf{PolyEnd} \ \xrightarrow{\ F\ } \ \textbf{PolyEnd} \ \xrightarrow{\ N_0\ } \ \textbf{PrSh}(\textbf{tEmb}).
$$

We will show it is a pointwise extension, i.e. that for each $\mathsf{P} \in \textbf{PolyEnd}$, the left Kan extension

$$
\begin{array}{ccc}
\textbf{tEmb}/\mathsf{P} & \longrightarrow & 1 \\
\downarrow & \overset{\lambda}{\Rightarrow} & \downarrow {\scriptstyle \ulcorner P \urcorner} \\
\textbf{tEmb} & \longrightarrow & \textbf{PolyEnd}
\end{array}
$$

is preserved by $N_0 \circ F$.

The claim is that (for fixed $X \in \textbf{PrSh}(\textbf{tEmb})$) each natural transformation

$$
\begin{array}{ccc}
\textbf{tEmb}/\mathsf{P} & \longrightarrow & 1 \\
\downarrow & \overset{\phi}{\Rightarrow} & \searrow {\scriptstyle \ulcorner X \urcorner} \\
\textbf{tEmb} \longrightarrow \textbf{PolyEnd} & \xrightarrow{F} \textbf{PolyEnd} & \xrightarrow{N_0} \textbf{PrSh}(\textbf{tEmb})
\end{array}
$$

factors uniquely as

$$\begin{array}{ccc}
\textit{tEmb}\,/\mathsf{P} & \longrightarrow & 1 \\
\downarrow & \overset{\lambda}{\Rightarrow} \quad \ulcorner\mathsf{P}\urcorner\downarrow & \overset{}{\underset{\psi}{\Rightarrow}} \quad \overset{\ulcorner X\urcorner}{\searrow} \\
\textit{tEmb} & \longrightarrow \textit{PolyEnd} \underset{F}{\longrightarrow} \textit{PolyEnd} \underset{N_0}{\longrightarrow} & \textit{PrSh}\,(\textit{tEmb}).
\end{array}$$

The component of ϕ at a P-tree $a : \mathsf{A} \to \mathsf{P}$ is a map of presheaves $\phi_a : \overline{\mathsf{A}} \to X$, i.e. for each (abstract) tree T a natural map

$$\phi_{a,\mathsf{T}} : \textit{PolyEnd}\,(\mathsf{T},\overline{\mathsf{A}}) \to X(\mathsf{T}).$$

To specify ψ we need for each tree T a natural map

$$\psi_{\mathsf{T}} : \textit{PolyEnd}\,(\mathsf{T},\overline{\mathsf{P}}) \to X(\mathsf{T}).$$

Finally, the component of $N_0 \circ F \circ \lambda$ at a P-tree $a : \mathsf{A} \to \mathsf{P}$ and an abstract tree T is

$$\begin{array}{ccc}
\textit{PolyEnd}\,(\mathsf{T},\overline{\mathsf{A}}) & \longrightarrow & \textit{PolyEnd}\,(\mathsf{T},\overline{\mathsf{P}}) \\
z & \longmapsto & \overline{a} \circ z.
\end{array}$$

Now the key point is that every $f \in \textit{PolyEnd}\,(\mathsf{T},\overline{\mathsf{P}})$ is in the image of this map for a suitable a: factor f into generic followed by free:

$$\begin{array}{ccc}
\mathsf{T} & \overset{f}{\longrightarrow} & \overline{\mathsf{P}} \\
{}_{g}\searrow & & \nearrow_{\overline{m}} \\
& \overline{\mathsf{M}} &
\end{array}$$

then

$$f = (N_0 \circ F \circ \lambda)_{m,\mathsf{T}}(g).$$

So if ψ is going to give ϕ after pasting with λ we are forced to define

$$\psi_{\mathsf{T}}(f) := \phi_{m,\mathsf{T}}(g) \in X(\mathsf{T}).$$

It is a routine calculation to verify that this assignment is natural in T. It relies on two facts: first, that generic-free factorisation is functorial: given $\mathsf{T}' \to \mathsf{T}$ over $\overline{\mathsf{P}}$ there is induced a unique $\mathsf{M}' \to \mathsf{M}$ between the factorisations, and second, that ϕ is natural in M. $\qquad\square$

2.2.11 Remark. The above arguments are analogous to those of Weber [21], Prop.4.22, and they serve in fact to prove the following general result: *let F be a monad on an arbitrary category $\mathscr{C}$, and let $i_0 : \Theta_0 \to \mathscr{C}$ be fully faithful and dense (with Θ_0 small). If the morphism $i_0{\downarrow}Fi_0{\downarrow}F \longrightarrow i_0{\downarrow}F$ of categories fibred over Θ_0 has connected fibres and admits a section (over Θ_0) then Θ_0 provides F with arities.*

With Proposition 2.2.10 we are in position to apply Weber's general nerve theorem (2.0.5) directly, establishing this:

2.2.12 Theorem. *The nerve functor* N : **PolyMnd** → **PrSh**(**tree**) *is fully faithful. A presheaf* X : **tree**^{op} → **Set** *is isomorphic to the nerve of a polynomial monad if and only if its restriction* j^*X : **tEmb**^{op} → **Set** *is isomorphic to the nerve of a polynomial endofunctor.* □

2.3   Sheaf conditions and nerve theorem for slices

Since we have now characterised polynomial monads in terms of their underlying endofunctors, we should now proceed to characterise polynomial endofunctors among all presheaves on **tEmb**.

Since every object in **PolyEnd** is a colimit of elementary trees (in a canonical way), the embedding **elTr** → **PolyEnd** is dense, and therefore also **tEmb** → **PolyEnd** is dense. This means that the nerve functor

$$N_0 : \textbf{PolyEnd} \longrightarrow \textbf{PrSh}(\textbf{tEmb})$$
$$\mathsf{P} \longmapsto \mathrm{Hom}_{\textbf{PolyEnd}}(-, \mathsf{P})$$

is fully faithful.

2.3.1. **Grothendieck topology on *TEmb*.** There is a Grothendieck topology on **TEmb** (and on **tEmb**): a family of tree embeddings $\{\mathsf{S}_\lambda \to \mathsf{T}\}_{\lambda \in \Lambda}$ is declared covering if it is jointly surjective on nodes, and also on edges. This topology has a more conceptual characterisation: the inclusion functor **elTr** → **tEmb** induces a geometric morphism

$$\textbf{PrSh}(\textbf{elTr}) \to \textbf{PrSh}(\textbf{tEmb})$$

which turns out to be a left exact localisation, hence defines a Grothendieck topology on **tEmb** — which is the one just described — inducing an equivalence

$$\textbf{PrSh}(\textbf{elTr}) \xrightarrow{\sim} \textbf{Sh}(\textbf{tEmb}).$$

2.3.2. **Reduced covers and generic injections.** A covering family on a tree T is called *reduced* if each node of T is only in one member of the family and if no member can be removed without spoiling the covering property. Another characterisation of reduced covers is: each outer edge of T is hit exactly once, and each inner edge is hit either once or twice. The reduced covers of T form a poset (**RedCov**(T), ≤) with $F \leq G$ if the cover F is a refinement of G. If T is a nontrivial tree, there are isomorphisms of posets

$$\textbf{RedCov}(\mathsf{T})^{op} \simeq \mathscr{P}(T^1 \times_{T^0} T^2) \simeq \textbf{GenInj}(\mathsf{T}).$$

Here the middle poset is the powerset of the set of inner edges in T, and **GenInj**(T) denotes the poset of isomorphism classes of generic injections into T. The left-hand correspondence is: for a subset J of inner edges, the reduced cover is given by cutting

the tree at the inner edges in *J*. In other words, the inner edges in *J* are those inner edges hit twice by the cover. For the right-hand correspondence, the inner edges in *J* correspond to the inner edges that are hit by a generic injection. In the correspondence between reduced covers and generic injections, the covering condition corresponds to the generic condition (boundary preservation), while the reducedness of the cover corresponds to injectivity of the generic map.

2.3.3. **The Segal condition.** Each tree is canonically the colimit of its elements, i.e. the colimiting cone of its canonical diagram, cf. 2.1.7. A presheaf $X : \textbf{\textit{tree}}^{\mathrm{op}} \to \textbf{\textit{Set}}$ is said to satisfy the *Segal condition* if X sends these cocones in **tree** to limit cones in **Set**. The cocones in turn are just iterated pushouts of grafting type, so we can also state the Segal condition as the requirement of sending those pushouts to pullbacks. Finally, the canonical diagrams correspond to the minimal covering families (those for which all members are elementary trees), and since these form a basis for the topology, we can also say that X satisfies the Segal condition if and only if it its restriction to **tEmb** is a sheaf.

2.3.4 Proposition. *If* P *is a polynomial endofunctor, then* $N_0\mathsf{P}$ *is a sheaf.*

Proof. This follows directly from the fact that the pushouts of 1.1.19 are also pushouts in the category **PolyEnd**. (Given $\mathsf{T} = \mathsf{S} +_{\mathsf{I}} \mathsf{R}$, then

$$\textbf{\textit{PolyEnd}}(\mathsf{T},\mathsf{P}) \longrightarrow \textbf{\textit{PolyEnd}}(\mathsf{S},\mathsf{P}) \times_{\textbf{\textit{PolyEnd}}(\mathsf{I},\mathsf{P})} \textbf{\textit{PolyEnd}}(\mathsf{R},\mathsf{P})$$

is an isomorphism by the pushout property.) ☐

With this result, we have factored the nerve functor N_0 as

$$\textbf{\textit{PolyEnd}} \xrightarrow{\quad N_0 \quad} \textbf{\textit{PrSh}}(\textbf{\textit{tEmb}})$$
$$\textbf{\textit{PrSh}}(\textbf{\textit{elTr}}) \simeq \textbf{\textit{Sh}}(\textbf{\textit{tEmb}})$$

and reduced the question to that of characterising polynomial endofunctors among presheaves on **elTr**. Before dealing with this (in the next section), a remark is due on the sliced case.

2.3.5. **Nerve theorem for slices.** There is a pushout theorem for **TEmb**/P and **PolyEnd**/P in analogy with 1.1.19, and there is induced a Grothendieck topology on **TEmb**/P giving an equivalence of categories

$$\textbf{\textit{PrSh}}(\textbf{\textit{elTr}}/\mathsf{P}) \simeq \textbf{\textit{Sh}}(\textbf{\textit{tEmb}}/\mathsf{P}).$$

Let now P be a polynomial monad. Composition of functors makes **PolyEnd**/P a 2-category; its monads are naturally identified with monads over P. In fact, **PolyEnd**/P

is the category of P-collections, and its monads are the P-operads, in the sense of Leinster [14, §4.2]. Again, the forgetful functor **PolyMnd** /P → **PolyEnd** /P has a left adjoint which can be described in terms of maps from P-trees in analogy with 1.2.8. This yields the *free* P-*monad monad* which is in fact cartesian. (For fixed set of objects, this is proved in Leinster's book [14, §C.1].) Furthermore, this monad is a local right adjoint, as it follows from the arguments in 2.2.6. (Since **PolyEnd** /P is a presheaf category the notion of local right adjoint is equivalent to the notion of familially representable, and the result can also be extracted from [14, §C.3].)

We have seen that every element n → P factors through a tree, and that all P-trees arise like this. The following result is now a direct application of the special nerve theorem (2.0.3).

2.3.6 Theorem. *For a presheaf* $X : (\textbf{tree}/\mathsf{P})^{\mathrm{op}} \to \textbf{Set}$*, the following are equivalent:*

1. *X is in the essential image of N (i.e. X is isomorphic to the nerve of a polynomial monad over* P*).*

2. *$j^{*}X$ is in the essential image of N_0 (i.e. $j^{*}X$ is isomorphic to the nerve of a polynomial endofunctor over* P*).*

3. *$j^{*}X$ is a sheaf on* $\textbf{tEmb}/\mathsf{P} \simeq \mathrm{tr}(\mathsf{P})$*.*

4. *X satisfies the Segal condition, i.e. takes the canonical cocones to limit cones.*

The key point to note is that we have an equivalence of categories

$$\textbf{PolyEnd}/\mathsf{P} \simeq \textbf{Sh}(\mathrm{tr}(\mathsf{P})).$$

2.3.7. **Examples.** If P is the identity monad, we recover the classical nerve theorem for categories. For P = M (the free-monoid monad), **tree** /M is the category of planar trees, polynomial monads over M are nonsymmetric operads, and the theorem says that such are characterised among presheaves on **tree** /M by the Segal condition.

2.4   Polynomial endofunctors and collections

2.4.1. **Collections.** The category **PrSh** (**elTr**) is the category of (coloured, symmetric) collections, which we denote by **Coll**. To be explicit, a collection C consists of a set $C(\iota)$ of *colours* and for each $n \in \mathbb{N}$ a set $C(n)$ of *n-ary operations*. The structure maps are first of all $n+1$ projections $C(n) \to C(\iota)$, and for each $n \in \mathbb{N}$ the symmetric group $\mathfrak{S}_n$ acts on $C(n)$ by permuting the first n projections. The inverse image in $C(n)$ of the elements $(i_1, \ldots, i_n; i) \in C(\iota)^{n+1}$ is denoted $C(i_1, \ldots, i_n; i)$ and is interpreted as the set of *n*-ary operations with input colours $i_1, \ldots, i_n$ and output colour i.

Since **elTr** is dense in **PolyEnd**, the nerve functor

$$
\begin{aligned}
R_0 : \textbf{PolyEnd} &\longrightarrow \textbf{PrSh}(\textbf{elTr}) = \textbf{Coll} \\
\mathsf{P} &\longmapsto \mathrm{Hom}(-, \mathsf{P})
\end{aligned}
$$

is fully faithful. We proceed to characterise its image, and start by looking at the slices:

2.4.2 Proposition. *The nerve functor R_0 : **PolyEnd** $\to$ **PrSh**(**elTr**) is a local equivalence. That is, for every polynomial endofunctor* P, *the sliced functor*

$$
\begin{aligned}
\mathbf{PolyEnd}/\mathsf{P} &\longrightarrow \mathbf{PrSh}(\mathbf{elTr})/R_0\mathsf{P} \\
[\mathsf{Q} \to \mathsf{P}] &\longmapsto [R_0\mathsf{Q} \to R_0\mathsf{P}]
\end{aligned}
$$

is an equivalence.

Proof. We use the equivalence $\mathbf{PolyEnd}/\mathsf{P} \simeq \mathbf{PrSh}(\mathrm{el}(\mathsf{P})) \simeq \mathbf{PrSh}(\mathbf{elTr}/\mathsf{P})$ of Proposition 2.1.3 (with equation (13)). Under this equivalence the sliced nerve functor has the following description: it sends a presheaf $\widetilde{Q} : (\mathbf{elTr}/\mathsf{P})^{\mathrm{op}} \to \mathbf{Set}$ to

$$
\begin{aligned}
X : \mathbf{elTr}^{\mathrm{op}} &\longrightarrow \mathbf{Set} \\
\mathsf{n} &\longmapsto \sum_{s:\mathsf{n}\to\mathsf{P}} \widetilde{Q}(s)
\end{aligned}
$$

(This presheaf has a natural map to $R_0\mathsf{P} = [\mathsf{n} \mapsto \mathrm{Hom}(\mathsf{n}, \mathsf{P})]$ by returning the running index of the sum.)

In the other direction, given a presheaf $X : \mathbf{elTr}^{\mathrm{op}} \to \mathbf{Set}$ with a map $\alpha : X \Rightarrow R_0\mathsf{P}$, define

$$
\begin{aligned}
\widetilde{Q} : (\mathbf{elTr}/\mathsf{P})^{\mathrm{op}} &\longrightarrow \mathbf{Set} \\
[s : \mathsf{n} \to \mathsf{P}] &\longmapsto X(\mathsf{n})_s
\end{aligned}
$$

where $X(\mathsf{n})_s$ denotes the fibre of $\alpha_\mathsf{n} : X(\mathsf{n}) \to \mathrm{Hom}(\mathsf{n}, \mathsf{P})$ over s.

It is easy to see that these two functors are inverse to each other, establishing the asserted equivalence. $\square$

2.4.3. **The nerve and its slices.** For each polynomial endofunctor P we have a diagram

$$
\begin{array}{ccc}
\mathbf{PolyEnd} & \xrightarrow{\ R_0\ } & \mathbf{PrSh}(\mathbf{elTr}) \\
\uparrow{\scriptstyle\dashv}\downarrow & & \uparrow{\scriptstyle\dashv}\downarrow \\
\mathbf{PolyEnd}/\mathsf{P} & \xrightarrow{\ \sim\ } & \mathbf{PrSh}(\mathbf{elTr})/R_0\mathsf{P}.
\end{array}
$$

The left adjoints are just forgetting the structure map to P and $R_0\mathsf{P}$, respectively, and clearly the square with the left adjoints commutes. The right adjoints are multiplication with P and multiplication with $R_0\mathsf{P}$, respectively. The square with the right adjoints commutes because R_0, as every nerve functor, commutes with limits, and in particular with products.

The right adjoint on presheaves has another conceptual description, via the equivalence $\mathbf{PrSh}(\mathbf{elTr})/R_0\mathsf{P} \simeq \mathbf{PrSh}(\mathbf{elTr}/\mathsf{P})$: in terms of the latter it is just precomposition with the forgetful functor $\mathbf{elTr}/\mathsf{P} \to \mathbf{elTr}$.

2.4.4. **Nonsymmetric collections.** The above diagram is most interesting when P is the free-monoid monad M: in this case we have

$$\mathbf{PrSh}(\mathbf{elTr})/R_0\mathsf{M} \simeq \mathbf{PrSh}(\mathrm{el}(\mathsf{M})),$$

and the latter is the category of *nonsymmetric collections*, denoted **NonSymColl**, and the left adjoint is then the symmetrisation functor, denoted S. The category $\mathrm{el}(\mathsf{M})$ is equivalent to the category of planar elementary trees. Nonsymmetric collections are described just as collections, except that there are no symmetries.

The diagram now reads:

$$
\begin{array}{ccc}
\mathbf{PolyEnd} & \xrightarrow{\makebox[3cm]{}R_0} & \mathbf{PrSh}(\mathbf{elTr}) = \mathbf{Coll} \\
L \dashv \updownarrow & & S \dashv \updownarrow \\
\mathbf{PolyEnd}/\mathsf{M} & \xrightarrow{\sim} \mathbf{PrSh}(\mathbf{elTr})/R_0\mathsf{M} \simeq & \mathbf{NonSymColl}.
\end{array}
$$

Note that L is surjective on objects. Indeed, every polynomial endofunctor admits a map to M. It follows that R_0 and S have the same essential image. Since R_0 is fully faithful we get:

2.4.5 Theorem. *The category **PolyEnd**, as a subcategory of **Coll**, is naturally identified with the Kleisli category for the symmetrisation monad S on **NonSymColl**.* □

2.4.6 Remark. The symmetrisation monad S on $\mathbf{PrSh}(\mathbf{elTr}/\mathsf{M})$ is a local right adjoint, since it is the composite of a forgetful functor from a slice category and a true right adjoint. It is endowed with arities by the representables themselves, and **elTr** appears as the Kleisli category of **elTr**/M with respect to S. The generic/free factorisation on **elTr** is quite degenerate: every arrow in **elTr** already *is* either a generic map (an automorphism of some n) or it is free (an inclusion ı → n).

Slightly more interesting is the corresponding generic/free factorisation system on **Tree**, the Kleisli category on the category **Tree**/M, still with respect to S. To see it most clearly, let **ptree** denote a skeleton of **Tree**/M, the category of planar trees. Let **tree**′ denote the full subcategory of **Tree** with one object for each object in **ptree** (i.e. the category of planar trees and not-necessarily planar maps). This is just the Kleisli category of S restricted to **ptree**. Now in **tree**′, the generics are the isomorphisms and the free maps are the planar maps, and every arrow factors as an isomorphism followed by a planar map (in analogy with the skeleton of the category of finite sets consisting of the sets $\{0, \ldots, n-1\}$ which happen to have a natural order: every arrow in this category factors as an isomorphism followed by an order-preserving map).

2.4.7. **Polynomial endofunctors as flat collections.** There is another characterisation of **PolyEnd** as a subcategory of **Coll**, suggested by André Joyal: it is the subcategory of 'projective' objects with respect to a certain class of surjections. Call a collection P *flat* if every colour-preserving surjection $Z \to P$ admits a section.

2.4.8 Theorem. *A collection P is isomorphic to the nerve of a polynomial endofunctor if and only if it is flat.*

Theorem 2.4.8 will be broken into a chain of biimplications (Theorem 2.4.10) each of which is rather easy to establish, once the correct viewpoint has been set up.

2.4.9. **Collections with a fixed set of colours.** We denote by $\textbf{\textit{Coll}}(I)$ the category of collections with colour set I and colour preserving morphisms. This category is again (equivalent to) a presheaf category. Namely, let $\textbf{\textit{MonEnd}}(I)$ be the full subcategory of $\textbf{\textit{PolyEnd}}(I)$ consisting of the *monomial* endofunctors (with endpoints I), i.e. those $I \leftarrow E \rightarrow B \rightarrow I$ for which B is singleton. Clearly $\textbf{\textit{MonEnd}}(I)$ is a groupoid. Write n for a fixed n-element set, and denote by $\textbf{\textit{monEnd}}(I) \subset \textbf{\textit{MonEnd}}(I)$ the small subgroupoid consisting of the objects

$$I \leftarrow n \rightarrow 1 \rightarrow I$$

($n \in \mathbb{N}$). It is a disjoint union:

$$\textbf{\textit{monEnd}}(I) = \sum_{n \in \mathbb{N}} \textbf{\textit{monEnd}}(I)_n$$

where $\textbf{\textit{monEnd}}(I)_n$ is the subgroupoid of monomials of degree n. Now we have

$$\textbf{\textit{Coll}}(I) \simeq \textbf{\textit{PrSh}}(\textbf{\textit{monEnd}}(I)).$$

(This viewpoint on $\textbf{\textit{Coll}}(I)$ was also used in the appendix of Berger-Moerdijk [4], except that they did not formulate it in terms of monomial functors.) An object $I \leftarrow n \rightarrow 1 \rightarrow I$ amounts to an $(n+1)$-tuple $(i_1, \ldots, i_n; i)$ of elements in I, and the value on it of a presheaf C is the set $C(i_1, \ldots, i_n; i)$; the arrows in $\textbf{\textit{monEnd}}(I)$ provide the colour-preserving symmetries. More formally, given a presheaf $F : \textbf{\textit{monEnd}}(I)^{\text{op}} \rightarrow \textbf{\textit{Set}}$, define a presheaf with colour set I by

$$
\begin{array}{ccc}
\textbf{\textit{elTr}}^{\text{op}} & \longrightarrow & \textbf{\textit{Set}} \\
\mathsf{n} & \longmapsto & \displaystyle\sum_{\mathsf{Q} \in \textbf{\textit{monEnd}}(I)_n} F(\mathsf{Q}),
\end{array}
$$

and conversely, given a presheaf $C : \textbf{\textit{elTr}}^{\text{op}} \rightarrow \textbf{\textit{Set}}$ with $C(\iota) = I$, define a presheaf on $\textbf{\textit{monEnd}}(I)$ by sending an object $(i_1, \ldots, i_n; i)$ to the inverse image of this $(n+1)$-tuple under the structure map $C(n) \rightarrow I$.

Since $\textbf{\textit{monEnd}}(I)$ is a full subcategory of $\textbf{\textit{PolyEnd}}(I)$ we have a nerve functor

$$
\begin{array}{cccc}
R_{0(I)} : \textbf{\textit{PolyEnd}}(I) & \longrightarrow & \textbf{\textit{PrSh}}(\textbf{\textit{monEnd}}(I)) \\
\mathsf{P} & \longmapsto & \text{Hom}_{\textbf{\textit{PolyEnd}}(I)}(-, \mathsf{P}),
\end{array}
$$

and this nerve functor is compatible with the global nerve $R_0 : \textbf{\textit{PolyEnd}} \rightarrow \textbf{\textit{Coll}}$. Precisely, the diagram

$$\begin{array}{ccc}
\textbf{\textit{PolyEnd}} & \xrightarrow{R_0} & \textbf{\textit{PrSh}}(\textbf{\textit{elTr}}) \;=\; \textbf{\textit{Coll}} \\
\big\uparrow & & \big\uparrow \\
\textbf{\textit{PolyEnd}}(I) & \xrightarrow[R_{0(I)}]{} & \textbf{\textit{PrSh}}(\textbf{\textit{monEnd}}(I)) \;\simeq\; \textbf{\textit{Coll}}(I)
\end{array}$$

commutes up to a natural isomorphism, as is easy to check.

2.4.10 Theorem. *For any collection P with set of colours I, the following are equivalent.*

1. *P is flat (i.e. every colour-preserving surjection onto P in* **Coll** *admits a section).*

2. *P is projective in* **Coll**(I) *(i.e. every surjection onto P in* **Coll**(I) *admits a section).*

3. *P is a sum of representables in* **Coll**$(I) \simeq$ **PrSh**(**monEnd**(I))*.*

4. *P is in the essential image of the nerve functor $R_{0(I)}$:* **PolyEnd**$(I) \to$ **Coll**(I)*.*

5. *P is in the essential image of the nerve functor R_0 :* **PolyEnd** $\to$ **Coll***.*

Proof. $(1) \Leftrightarrow (2)$: obvious.

$(2) \Leftrightarrow (3)$: It is true in any category of presheaves on a groupoid that an object is projective (with respect to termwise surjections) if and only if it is a sum of representables.

$(3) \Leftrightarrow (4)$: The $R_{0(I)}$-nerve of a polynomial functor $I \leftarrow E \to B \to I$ is the sum of the representables $I \leftarrow E_b \to \{b\} \to I$ (indexed by $b \in B$). Conversely any sum of representables defines a polynomial endofunctor (with endpoints I). Phrased more elegantly: polynomial endofunctors are precisely the sums of monomial endofunctors.

$(4) \Leftrightarrow (5)$: This follows from the fact that the nerve functors are compatible. $\qquad\square$

2.5 Polynomial monads and operads

In this final section we characterise the image of the nerve functor for polynomial monads, explain the relation with operads, and sum up the relation between the various nerve functors.

Combining Proposition 2.2.12 and Proposition 2.4.8, we already have:

2.5.1 Proposition. *A presheaf X :* **tree**$^{\mathrm{op}} \to$ **Set** *is isomorphic to the nerve of a polynomial monad if and only if j^*X :* **tEmb**$^{\mathrm{op}} \to$ **Set** *is a flat collection.*

We now finally come to operads, and aim to fit everything into this diagram:

$$
\begin{array}{ccccccc}
N: & \textbf{PolyMnd} & \xrightarrow{\;R\;} & \textbf{Opd} & \xrightarrow{\;W\;} & \textbf{PrSh}(\textbf{tree}) \\[2ex]
   & \dashv\;\big\updownarrow & & \dashv\;\big\updownarrow & & \dashv\;\big\updownarrow\; j^* \\[2ex]
N_0: & \textbf{PolyEnd} & \xrightarrow{\;R_0\;} & \textbf{Coll} & \xrightarrow{\;W_0\;} & \textbf{PrSh}(\textbf{tEmb}) \\[2ex]
   & \dashv\;\big\updownarrow & & \dashv\;\big\updownarrow & & \dashv\;\big\updownarrow\; k^* \\[2ex]
N_0^{\mathrm{pl}}: & \textbf{PolyEnd}/\mathrm{M} & \xrightarrow[R_0^{\mathrm{pl}}]{\;\simeq\;} & \textbf{NonSymColl} & \xrightarrow[W_0^{\mathrm{pl}}]{} & \textbf{PrSh}(\textbf{tEmb}/\mathrm{M})
\end{array}
$$

The middle row and the bottom row were explained in the previous section.

2.5.2. **Operads.** For fixed set of colours I, the category $\textbf{Coll}(I)$ has a monoidal structure given by the substitutional tensor product (cf. [11], see also the Appendix of [4]). The monoids in $\textbf{Coll}(I)$ are the *I-coloured operads*, forming a category $\textbf{Opd}(I)$ and fitting into a free-forgetful adjunction $\textbf{Opd}(I) \rightleftarrows \textbf{Coll}(I)$. Morphisms between operads of different colours can be defined in terms of base-change (just as for collections): if $\alpha : I \to J$ is a map of sets there is an obvious change-of-colours functor $\alpha^* : \textbf{Opd}(J) \to \textbf{Opd}(I)$, and a morphism from an I-coloured operad X to a J-coloured operad Y is defined to be a pair (α, ϕ) where $\alpha : I \to J$ is a map of sets and $\phi : X \to \alpha^* Y$ is a morphism of I-coloured operads. With this extra structure the free-forgetful adjunctions assemble into a single adjunction

$$\textbf{Opd} \rightleftarrows \textbf{Coll}.$$

The functor $R_0 : \textbf{PolyEnd} \to \textbf{Coll}$ is monoidal (for each I) and commutes with base-change, hence induces the functor

$$R : \textbf{PolyMnd} \to \textbf{Opd}.$$

2.5.3. **The nerve functor for operads.** The nerve functor $W : \textbf{Opd} \to \textbf{PrSh}(\textbf{tree})$ is now defined in the obvious way from the embedding $\textbf{tree} \subset \textbf{PolyMnd} \subset \textbf{Opd}$. It is the nerve functor for operads introduced by Moerdijk and Weiss. The image of W is characterised in the following theorem, due to Moerdijk-Weiss and Weber:

2.5.4 Theorem. *For a presheaf $X : \textbf{tree}^{\mathrm{op}} \to \textbf{Set}$, the following are equivalent.*

1. *X is isomorphic to the nerve of an operad (i.e. X is in the essential image of W).*

2. *Every inner horn of X has a unique filler.*

3. *$j^* X$ is isomorphic to the nerve of a collection (i.e. $j^* X$ is in the essential image of W_0).*

4. *$k^* j^* X$ is isomorphic to the nerve of a nonsymmetric collection (i.e. $k^* j^* X$ is in the essential image of W_0^{pl}).*

5. *X satisfies the Segal condition (i.e. takes canonical cocones to limit cones.)*

Proof. The equivalence of (1) and (2) is due to Moerdijk and Weiss [18] (their Proposition 5.3 together with Theorem 6.1). We shall not need the horn-filling condition here, and mention the result only because it was the first nerve theorem for operads. The equivalence of (1), (4) and (5) are due to Weber [21] (combining his Examples 2.14, 4.19, and 4.27). Condition (3) and the category **tEmb**, central to the present paper, are not considered by Moerdijk and Weiss, nor by Weber. The equivalence between (4) and (3) follows from the fact that the symmetrisation functor is a local right adjoint on a presheaf category (cf. 2.4.6). ☐

For emphasis, we restate Proposition 2.2.12 as a characterisation of polynomial monads among operads:

2.5.5 Proposition. *An operad is isomorphic to a polynomial monad if and only if its underlying collection is flat.*

☐

References

[1] JOHN C. BAEZ and JAMES DOLAN. *Higher-dimensional algebra. III. n-categories and the algebra of opetopes.* Adv. Math. **135** (1998), 145–206. ArXiv:q-alg/9702014.

[2] MICHAEL BARR and CHARLES WELLS. *Toposes, triples and theories*. No. 278 in Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, 1985. Corrected reprint in Reprints in Theory and Applications of Categories, 12 (2005) (electronic).

[3] CLEMENS BERGER. *A cellular nerve for higher categories*. Adv. Math. **169** (2002), 118–175.

[4] CLEMENS BERGER and IEKE MOERDIJK. *Resolution of coloured operads and rectification of homotopy algebras*. In *Categories in algebra, geometry and mathematical physics*, vol. 431 of Contemp. Math., pp. 31–58. Amer. Math. Soc., Providence, RI, 2007. ArXiv:math/0512576.

[5] NICOLA GAMBINO and MARTIN HYLAND. *Wellfounded Trees and Dependent Polynomial Functors*. In S. Berardi, M. Coppo, and F. Damiani, editors, *TYPES 2003*, vol. 3085 of Lecture Notes in Computer Science, pp. 210–225. Springer Verlag, Heidelberg, 2004.

[6] NICOLA GAMBINO and JOACHIM KOCK. Polynomial functors and polynomial monads. Preprint, arXiv:0906.4931.

[7] VICTOR GINZBURG and MIKHAIL KAPRANOV. *Koszul duality for operads*. Duke Math. J. **76** (1994), 203–272. ArXiv:0709.1228.

[8] ANDRÉ JOYAL. *Foncteurs analytiques et espèces de structures*. In *Combinatoire énumérative (Montréal/Québec, 1985)*, vol. 1234 of Lecture Notes in Math., pp. 126–159. Springer, Berlin, 1986.

[9] ANDRÉ JOYAL. *Disks, duality and Θ-categories*. Preprint, September 1997.

[10] ANDRÉ JOYAL and JOACHIM KOCK. *Feynman graphs, and nerve theorem for compact symmetric multicategories (extended abstract)*. To appear in the proceedings of "Quantum Physics and Logic VI", Oxford, 2009. Electronic Notes in Theoretical Computer Science. ArXiv:0908.2675.

[11] G. MAX KELLY. *On the operads of J. P. May*. Repr. Theory Appl. Categ. (2005), 1–13 (electronic). Written 1972.

[12] JOACHIM KOCK. Notes on polynomial functors. Rough draft, 420pp. Available from http://mat.uab.cat/~kock/cats/polynomial.html.

[13] JOACHIM KOCK, ANDRÉ JOYAL, MICHAEL BATANIN, and JEAN-FRANÇOIS MASCARI. *Polynomial functors and opetopes*. Preprint, arXiv:0706.1033.

[14] TOM LEINSTER. *Higher Operads, Higher Categories*. London Math. Soc. Lecture Note Series. Cambridge University Press, Cambridge, 2004. ArXiv:math.CT/0305049.

[15] TOM LEINSTER. Nerves of algebras. Talk at CT04, Vancouver. Slides available from http://www.maths.gla.ac.uk/~tl/vancouver/.

[16] IEKE MOERDIJK and ERIK PALMGREN. *Wellfounded trees in categories*. Annals of Pure and Applied Logic **104** (2000), 189–218.

[17] IEKE MOERDIJK and ITTAY WEISS. *Dendroidal sets*. Alg. Geom. Top. (2007), 1441–1470. ArXiv:math/0701293.

[18] IEKE MOERDIJK and ITTAY WEISS. *On inner Kan complexes in the category of dendroidal sets*. Adv. Math. **221** (2009), 343–389. ArXiv:math/0701295.

[19] ROSS STREET. *The petit topos of globular sets*. J. Pure Appl. Algebra **154** (2000), 299–315. Category theory and its applications (Montreal, QC, 1997).

[20] MARK WEBER. *Generic morphisms, parametric representations and weakly Cartesian monads*. Theory Appl. Categ. **13** (2004), 191–234 (electronic).

[21] MARK WEBER. *Familial 2-functors and parametric right adjoints*. Theory Appl. Categ. **18** (2007), 665–732 (electronic).

Joachim Kock <`kock@mat.uab.cat`>
Departament de matemàtiques
Universitat Autònoma de Barcelona
08193 Bellaterra (Cerdanyola), Spain