Preliminaries

0.1. Category theory

Throughout this work we will largely ignore size issues regarding such problems as the existence of the category of all categories and similar questions. We will assume a suitable setting of universes and content ourselves with the construction of the category of all small sets, all small categories, all small spaces and so on, where 'small' is meant with respect to some universe of discourse. We will not always explicitly mention the word 'small', always assuming it is meant when necessary.

We start by setting the notation of category theory used in this work. Most of the category theory used herein is rather elementary and almost all of it can be found in **[34]**, to which the reader is referred to for more details if needed. Our choice of notation is somewhat different than the one presented in **[34]** but is still (largely) the standard one. Given a category $\mathcal{C}$ we denote its set of objects by $ob(\mathcal{C})$. For every two objects $a, b \in ob(\mathcal{C})$ we denote by $\mathcal{C}(a, b)$ the set of arrows with domain a and codomain b.

Given a symmetric monoidal category $\mathcal{C}$, it is said to be a *closed* monoidal category if for each $b \in ob(\mathcal{C})$ the functor $- \otimes b : \mathcal{C} \to \mathcal{C}$, that sends an object a to $a \otimes b$, has a right adjoint. Usually this right adjoint is called the *internal Hom* in $\mathcal{C}$ and is denoted by $\underline{Hom}_{\mathcal{C}}(b, -)$. We deviate from this notation and introduce the notation $\underline{\mathcal{C}}(b, -)$ instead. By definition then, there is an isomorphism

$$\mathcal{C}(a \otimes b, c) \cong \mathcal{C}(a, \underline{\mathcal{C}}(b, c))$$

natural in a, b, and c.

The monoidal functors that would interest us would always be the strong monoidal ones, i.e., those in which the components of the coherence natural transformations are isomorphisms. If the monoidal structure on a category is given by categorical products then we will call it a *cartesian* category. If that monoidal category is closed we will call it a *cartesian closed*.

Example 0.1.1. There are many examples of closed monoidal categories such as Cat with the cartesian product of categories, $Vect$ with the usual tensor product of vector spaces, etc. With our notation, given two categories $\mathcal{C}$ and $\mathcal{D}$ in $ob(Cat)$, the *set* $Cat(\mathcal{C}, \mathcal{D})$ is the set of all functors $F : \mathcal{C} \to \mathcal{D}$ while $\underline{Cat}(\mathcal{C}, \mathcal{D})$ is the *category* whose objects are all functors $F : \mathcal{C} \to \mathcal{D}$ and whose arrows are natural transformations between such functors.

0.1.1. Presheaf categories. We now introduce presheaf categories and mention some of their basic properties. Given a category Γ, the *presheaf* category on Γ is the category $Set^{\Gamma^{op}}$ of contravariant functors $\Gamma^{op} \to Set$ and their natural transformations. Given such a presheaf $X : \Gamma^{op} \to Set$ we write, for each object

$\gamma \in ob(\Gamma)$,

$$X_\gamma = X(\gamma).$$

For our purposes it will be convenient to think of presheaf categories as follows. Consider the category Γ as a category whose objects are shapes and the arrows express the way different shapes relate to one another. A presheaf on Γ should then be thought of as a set each of whose elements has a shape. In more detail, the presheaf $X : \Gamma^{op} \to Set$ should be thought of as a set where an element $x \in X_\gamma$ has shape γ. The contravariance of X as a functor means that an element $x \in X_\gamma$ of shape γ has other elements of other shapes associated to it. For example, if $\gamma' \to \gamma$ is a monomorphism in Γ (which can be interpreted as saying that γ' is a sub-shape of γ) then there is a function $X_\gamma \to X_{\gamma'}$ (which can be interpreted as mapping $x \in X_\gamma$ to an element $x' \in X_{\gamma'}$ where x' is the 'part' of x 'shaped' like γ').

A fundamental property of presheaf categories is that every presheaf is the canonical colimit of representable presheaves. In more detail, given a category Γ, each $\gamma \in ob(\Gamma)$ induces a *representable presheaf* denoted by $\Gamma[\gamma] = \Gamma(-, \gamma)$ and defined by

$$\Gamma[\gamma]_{\gamma'} = \Gamma(\gamma', \gamma).$$

Given an arbitrary presheaf $X : \Gamma^{op} \to Set$, consider all maps $\Gamma[\gamma] \to X$ for all possible $\gamma \in ob(\Gamma)$. By the Yoneda Lemma these correspond exactly to all $x \in X_\gamma$. Assigning $\Gamma[\gamma]$ to each $\Gamma[\gamma] \to X$ we obtain a diagram in $Set^{\Gamma^{op}}$. The colimit of this diagram is the original presheaf X. We write this shortly as

$$X = \varinjlim_{\Gamma[\gamma] \to X} \Gamma[\gamma].$$

Lastly we introduce a categorical construction (a special case of a Kan extension) which will repeatedly be used in this work. Given two categories $\mathcal{C}$ and $\mathcal{D}$ we denote by $adj(\mathcal{D}, \mathcal{C})$ the category of adjunctions between $\mathcal{D}$ and $\mathcal{C}$.

PROPOSITION 0.1.2. *Let $\mathcal{C}$ be a cocomplete category and Γ an arbitrary category. There is an equivalence of categories between the category of functors from Γ to $\mathcal{C}$ and the category of adjunctions between the categories $Set^{\Gamma^{op}}$ and $\mathcal{C}$, i.e,*

$$\underline{Cat}(\Gamma, \mathcal{C}) \simeq adj(Set^{\Gamma^{op}}, \mathcal{C}).$$

PROOF. We describe the effect on objects of two functors

$$\underline{Cat}(\Gamma, \mathcal{C}) \rightleftarrows adj(Set^{\Gamma^{op}}, \mathcal{C})$$

which together constitute the desired equivalence. We omit most of the details, which are just simple verifications.

Given a functor $F : \Gamma \to \mathcal{C}$ we need to construct an adjunction

$$Set^{\Gamma^{op}} \underset{N_F}{\overset{|-|_F}{\rightleftarrows}} \mathcal{C} .$$

The functor $N_F : \mathcal{C} \to Set^{\Gamma^{op}}$ is defined for an object $C \in ob(\mathcal{C})$ to be the presheaf $N_F(C)$ whose elements of shape γ form the set $N_F(C)_\gamma = \mathcal{C}(F(\gamma), C)$. To define the functor $|-|_F$ for an arbitrary presheaf $X : \Gamma^{op} \to Set$, recall the canonical presentation of X as a colimit of representables

$$X = \varinjlim_{\Gamma[\gamma] \to X} \Gamma[\gamma].$$

We then define

$$|X|_F = \varinjlim_{\Gamma[\gamma]\to X} F(\gamma),$$

which exists since $\mathcal{C}$ was assumed cocomplete. We now prove that $|-|_F$ is indeed the left adjoint of N_F. Given $X \in ob(Set^{\Gamma^{op}})$ and $C \in ob(\mathcal{C})$ we need to establish a natural isomorphism

$$Set^{\Gamma^{op}}(X, N_F(C)) \cong \mathcal{C}(|X|_F, C).$$

To simplify the notation we neglect the subscript F. We now obtain the desired natural isomorphism by the following calculation (all colimits and limits are taken over the same diagram as above):

$$\begin{aligned}
Set^{\Gamma^{op}}(X, N(C)) &\cong \\
Set^{\Gamma^{op}}(\varinjlim \Gamma[\gamma], N(C)) &\cong \\
\varprojlim Set^{\Gamma^{op}}(\Gamma[\gamma], N(C)) &\cong \\
\varprojlim N(C)_\gamma &\cong \\
\varprojlim \mathcal{C}(F\gamma, C) &\cong \\
\mathcal{C}(\varinjlim F\gamma, C) &\cong \\
\mathcal{C}(|X|, C).
\end{aligned}$$

To construct a functor $\Gamma \to C$ from a given adjunction $Set^{\Gamma^{op}} \underset{U}{\overset{G}{\rightleftarrows}} \mathcal{C}$ we simply define $F(\gamma) = G(\Gamma[\gamma])$. This completes the (sketch of the) proof. □

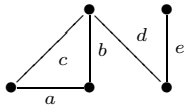
In the setting of the above construction we will refer to a functor $F : \Gamma \to \mathcal{C}$ as a *probe*. This is to be thought of as mapping each shape $\gamma \in ob(\Gamma)$ to an object $F(\gamma) \in ob(\mathcal{C})$ in a functorial way, thus specifying certain objects in $\mathcal{C}$ which behave somewhat like the shapes γ. By 'probing' an object $C \in ob(\mathcal{C})$, by mapping into it from the various objects $F(\gamma)$, we then obtain the presheaf $N_F(C)$. We call N_F the *nerve* functor induced by the probe F and the left adjoint $|-|_F$ the *realisation* in $\mathcal{C}$ of a presheaf $X : \Gamma^{op} \to Set$ induced by F. The realisation process uses the information in the presheaf X as instructions on how to 'glue' the various objects $F(\gamma)$ in $\mathcal{C}$.

0.2. A formalism of trees

Trees play a fundamental role in the theory of operads in general and in this work as well. We present here a formalism of trees which is somewhat different from the standard ones (see **[18, 35]** for two approaches to trees). Our trees are based on the notion of a graph. However, usually a graph is given by specifying a set of vertices V, and the edges are then a certain subset of $V \times V$. We will find it more convenient to have a definition of a graph where the basic ingredient is the set of edges, and the vertices are then defined in terms of those.

Definition 0.2.1. A *graph* G consists of a non-empty set E of *edges* and a set $V \subseteq P(E)$ of *vertices* such that every edge belongs to at most two different vertices. Those edges that belong to two distinct vertices are called *inner* while those that belong to less then two vertices are called *outer*.

We will draw graphs in the usual way. For example, the picture



denotes the graph whose set of edges is $\{a, b, c, d, e\}$ with the following vertices

$$\{\{a, c\}, \{a, b\}, \{c, b, d\}, \{d, e\}, \{e\}\}.$$

Notice that our definition excludes graphs with edges from a vertex to itself and also graphs with two vertices and several parallel edges between these two vertices, and similar graphs. This will not concern us since our main interest is trees, in which such graphs do not occur.

If two distinct edges e, e' in a graph belong to the same vertex we say that e and e' are *linked*. For a given edge e, if e belongs to two distinct vertices u and v then we say that u and v are *adjacent* and that e *connects* u and v.

DEFINITION 0.2.2. A *path* of length $n \geq 1$ in a graph is a sequence of edges

$$e_1, \cdots, e_n$$

such that e_i is linked to e_{i+1} for each $1 \leq i < n$. We say that two edges e, e' are *connected* if there is a path as above with $e_1 = e$ and $e_n = e'$. A *loop* is then a path $e_1, \cdots, e_n$ of length of at least 2 such that $e_1 = e_n$. A graph is said to be *connected* if any two edges in it are connected.

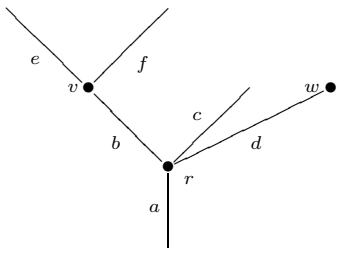
In a connected graph G we distinguish two kinds of vertices:

DEFINITION 0.2.3. Let G be a connected graph. A vertex v that consists of just one edge is called an *outer* vertex. The other vertices are called *inner* vertices.

0.2.1. Trees.

DEFINITION 0.2.4. A *tree* is a finite connected graph with no loops and a chosen outer edge called the *root*. The rest (if any) of the outer edges are called *leaves*.

We will draw trees with their root at the bottom and directed from the leaves to the root. We will use a $\bullet$ for vertices. The direction in the tree defines for each vertex v a unique outgoing edge denoted by $out(v)$ (called the *output* of the vertex) and a (possibly empty) set of incoming edges denoted by $in(v)$ (called the *input* of the vertex). The number of incoming edges into v is called the *valence* of v. For example in the tree:



there are three vertices of valence 2,3, and 0 and three leaves (at the outer sides of the edges e, f and c). The outer edges are e, f, c, and a, where a is the root. The inner edges are then b and d.

Thus a tree T is given by $(E(T), V(T), out(T))$ where $(E(T), V(T))$ is a finite, connected, loop-free graph and $out(T)$ is the chosen root. We will use the notation $in(T)$ to refer to the set of leaves of the tree.

The tree

consisting of just one edge and no vertices is called the *unit* tree We denote this tree by η, or η_e if we wish to explicitly name the unique edge. In this tree, its only edge is both the root and a leaf.

DEFINITION 0.2.5. Let T and S be two trees whose only common edge is the root r of S which is also one of the leaves of T. The *grafting*, $T \circ S$, of S on T along r is the graph

$$(E(T) \cup E(S), V(T) \cup V(S), out(T)).$$

That this indeed defines a tree is easily checked. Pictorially the grafted tree $T \circ S$ is obtained by putting the tree S on top of the tree T by identifying the output edge of S with the input edge r of T. By repeatedly grafting, one can define a full grafting operation $T \circ (S_1, \cdots, S_n)$ whenever the set of the roots of the trees S_i is equal to the set of leaves of T, the sets $E(S_i)$ are pairwise disjoint, and $E(S_i)$ meets $E(T)$ at a leaf of T which is also the root of S_i (for each $1 \leq i \leq n$).

We now state a useful decomposition of trees that allows for inductive proofs on trees. The proof is trivial.

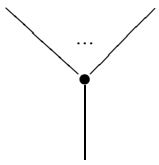
PROPOSITION 0.2.6. *(Fundamental decomposition of trees) Let T be a tree. Suppose T has root r and $\{r, e_1, \cdots, e_n\}$ is the vertex containing r. Let T_{e_i} be the tree that contains the edge e_i as root and everything above it in T. Then*

$$T = T_{root} \circ (T_{e_1}, \cdots, T_{e_n})$$

where T_{root} is the tree consisting of r as root and $\{e_1, \cdots, e_n\}$ as the set of leaves.

Below, certain trees will appear often. For easy reference we define them here.

DEFINITION 0.2.7. A tree C_n of the form:



that has just one vertex and n leaves will be called an *n-corolla* (or a *corolla* if we do not wish to specify the number of leaves). A tree of the form

with one leaf and only unary vertices will be called a *linear tree*. If the edges are numbered $0, \cdots, n$ from the root up, we denote it by L_n.

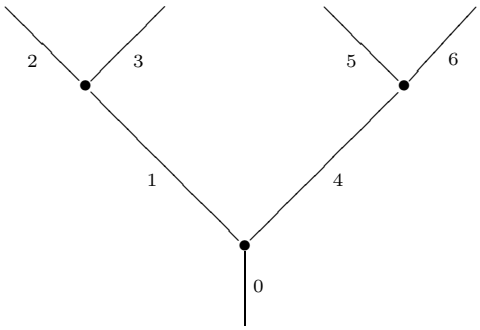
0.2.2. Planar trees.

DEFINITION 0.2.8. A *planar tree* $\bar{T}$, is a tree T together with a linear ordering of $in(v)$ for each vertex v.

The ordering of $in(v)$ for each vertex is equivalent to drawing the graph on the plane. It is evident that a single tree can become a planar tree in (usually) many different ways.

DEFINITION 0.2.9. Let T be a planar tree with n edges. We call T a *standard* planar tree if its set of edges is $E = \{0, 1, 2, \cdots, n\}$ and if when the tree is traversed left-first from the root up then the edges appear in the natural order on E.

EXAMPLE 0.2.10. The following is a standard planar tree:



It is obvious that any planar rooted tree is, up to a renaming of the edges, the same as precisely one of the standard planar trees.

The grafting of planar trees is defined just as that of non-planar ones and it is evident that the same fundamental decomposition property still holds for planar trees. We now define the grafting of standard planar trees. Let T and S be two standard planar trees. The leaves of T can be numbered from left to right. Let i be the i-th leaf. The grafting $T \circ_i S$ is given as follows. Rename the edges of S such that the root of S is equal to l, the i-th leaf of T, and such that $E(S') \cap E(T) = \{l\}$. Call the new tree S'. With this notation we have the following definition.

DEFINITION 0.2.11. The *grafting* of the standard planar trees S and T along the i-th leaf is denoted by $T \circ_i S$ and is the standard planar tree that has the same shape as the tree $T \circ_l S'$, obtained by ordinary grafting.