Finite Lawvere Theory

Jorian Pruim

March 31, 2025

Abstract

In this paper we construct a framework to study algebraic structures through the lens of a tweaked version of Lawvere Theory. We do this by encoding operations and properties of the structures in suitable templates in the language of categories. Furthermore we exhibit some examples and note some of the limitations of this construction.
In the first section we go over the established background definitions used in the language of categories, and in the second we provide a step-by-step motivation of the construction and properly define it.

Introduction

In 1963, William Lawvere published his Ph.D. thesis[Law63b][Law63a] where a framework for defining algebraic structures is presented by encoding all algebraic operations on the structure in a small category closed under all finite products where every object is isomorphic to a finite cartesian power x^n of a generic object x[nla24]. Many variations on this method of modelling algebraic structures have been put forth, like the theory of operads and its more generalized counterpart "colored operads" (equivalently, multicategories)[Lei04]; infinitary Lawvere theory, which lifts the demand for algebraic operations to have finite arity to allow for a supremum operation; and multisorted lawvere theory[Tri18], which allows for more than one base set to be present in the theory. The latter can be further generalized to the theory of sketches[Bra23], which allows for colimits to play a role in determining the strucutre's behaviour.
In this thesis we will describe a framework resembling multisorted lawvere theory from the ground up, whilst waiving the requirement for the aforementioned small category to be closed under finite products. Strictly, we consider categories, named template categories, in which each object is equal to a finite product of designated generic objects, which we will call "generating objects".
As such, we are able to fully represent an algebraic theory by a finite amount of objects and morphisms, hence the name Finite Lawvere Theory. Some things that are of interest are the algebras formed by a fixed template category T, given by product preserving functors, and the natural transformations between those functors. Combined they form a new category of finite lawvere algebras A_T. The naturality condition ensures that given a natural transformation φ any algebraic operation encoded by the template is preserved by the underlying functions φ_x on the sets associated with the generic objects x.
Another thing we touch upon are the morphisms between the template categories itself, also given by product preserving functors, and explaining how they allow the structure of one algebra to be imposed on the other, for example by showing that the category of groups embeds nicely into the category of pointed sets.
One main construction in the thesis is a recipe to create the template categories for algebraic structures with one underlying base set, by endowing a finite category with freely generated product morphisms and formal compositions and identifying the arrows corresponding to the sides of an equation in free variables present in the algebra. Afterwards we show that the category of product preserving functors from the template category into the category of sets is equivalent as a category to the category of the corresponding algebraic theory.
Lastly, we provide some examples and argue why the category of algebras stemming from a template category must be closed under finite products itself. This makes it difficult or sometimes impossible to model partial operations like taking inverses in rings, and as such the category of fields cannot be equivalent to a category of finite lawvere algebras.

1 Background Definitions on Categories

1.1 Categories and Functors

Category theory is a rich and relatively new subject that specialises in the study of how things relate to each other. Much is said about the subject and even more is still open to be discovered, but in this section we will discuss the basic notions that are used further along in the paper. For a more detailed text on categories as a whole we recommend [Lan78].

Definition 1. *A* ***category*** C *consists of a class*[1] *of objects, denoted by uppercase symbols, and for every two objects, not necessarily distinct, a set*[2] *of morphisms, denoted by lowercase symbols.*
The class of objects of a category is denoted $\mathsf{Ob}(\mathsf{C})$*, and the set of morphisms between two objects* $X, Y \in \mathsf{Ob}(\mathsf{C})$ *is denoted* $\mathsf{C}(X, Y)$*. Instead of writing* $f \in \mathsf{C}(X, Y)$*, the notation* $f : X \to Y$ *may be used when the category* C *is known from the context.*
For every three objects X, Y, Z*, there exists a composition function* $\circ : \mathsf{C}(Y, Z) \times \mathsf{C}(X, Y) \to \mathsf{C}(X, Z)$ *that is associative; meaning that for* $f \in \mathsf{C}(X, Y), g \in \mathsf{C}(Y, Z), h \in \mathsf{C}(Z, W)$ *it holds that* $h \circ (g \circ f) = (h \circ g) \circ f$*.*
For every object X*, the set* $\mathsf{C}(X, X)$ *contains an element* id_X *that is neutral under composition, that is, for all* $f \in \mathsf{C}(X, Y)$ *and* $g \in \mathsf{C}(Y, X)$ *one finds that* $f \circ id_X = f$ *and* $id_X \circ g = g$*.*

Notably, id_X is unique in the sense that every morphism that is neutral under composition must equal it; suppose id_X^1, id_X^2 are two such morphisms:

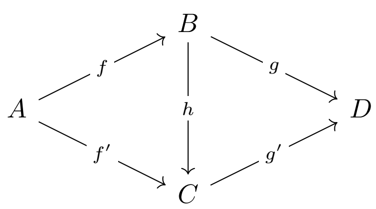
$$id_X^1 = id_X^1 \circ id_X^2 = id_X^2$$

where right neutrality and left neutrality are applied respectively.

Example 1. *Textbook examples of categories are:*

- *The category of sets, denoted* Set*, has objects are sets and has morphisms as functions between sets.*
- *Various algebraic structures, like groups, have associated categories, like* Grp*, whose morphisms are homomorphisms.*
- *For a fixed field* K*, one may consider the category of* K*-vector spaces, denoted* Vect_K*. The morphisms are linear transformations.*
- *Not all categories consists of sets with structure. Any set with a partial order*[3] $(X, \leq)$ *induces a category* X *where* $\mathsf{Ob}(\mathsf{X}) = X$ *and for every* $x, y \in X$*, the homset* $\mathsf{X}(x, y)$ *contains a unique morphism if* $x \leq y$ *and is empty otherwise. Composition in this category is defined in the only way possible, and is well-defined because of the transitive property of partial orderings. Identity morphisms are guaranteed by reflexiveness of the ordering.*
- *Notably, there exists a category of categories. The morphisms in this category are known as functors, which are defined in definition 3.*

Morphisms are the most important part of a category, and they almost never appear alone. Rather, it is of interest how the morphisms interact with each other. The use of diagrams helps to keep track of what morphisms are in play and how they relate to each other.



Definition 2. *A diagram is called* ***commutative*** *if all parallel morphisms obtained by composition within the diagram are equal. Intuitively, it means that traversing the diagram will yield the same result independent on the choice of path.*
Concretely, the diagram above commutes if $g \circ f = g' \circ f'$*,* $h \circ f = f'$*,* $g' \circ h = g$*.*

Functors are best seen as "homomorphisms between categories".

[1] As a precaution to not stumble over well-foundedness paradoxes in set theory and the foundations of mathematics, a class is to be seen as a collection of things. This way we can converse about the category of sets, whose object class is not a set itself. A more thorough definition of a class is unfortunately outside the scope of this discussion.

[2] Generally this is also taken to be a class, but to avoid complications and technicalities, we only consider this type of categories better known as socally small categories

[3] Read: a set together with a reflexive and transitive relation

Definition 3. *Let C, D be categories. A* ***functor*** *$F : \mathsf{C} \to \mathsf{D}$ consists of a map $F : \mathsf{Ob}(\mathsf{C}) \to \mathsf{Ob}(\mathsf{D})$ and for every two objects $X, Y \in \mathsf{Ob}(\mathsf{C})$ a function $F : \mathsf{C}(X, Y) \to \mathsf{D}(F(X), F(Y))$.*
For convenience and to prevent cluttering, most of the time the parentheses are ommitted and we write FX instead of $F(X)$. Sometimes the parentheses are reintroduced for clarity.
A functor is a structure preserving map between categories and is subject to the following conditions:

- *The functor preserves the identity map: $F(id_X) = id_{FX}$ for all $X \in \mathsf{Ob}(\mathsf{C})$.*
- *The functor preserves composition: $F(f \circ g) = Ff \circ Fg$ for any three objects $X, Y, Z \in \mathsf{Ob}(\mathsf{C})$ and morphisms $f \in \mathsf{C}(Y, Z), g \in \mathsf{C}(X, Y)$.*

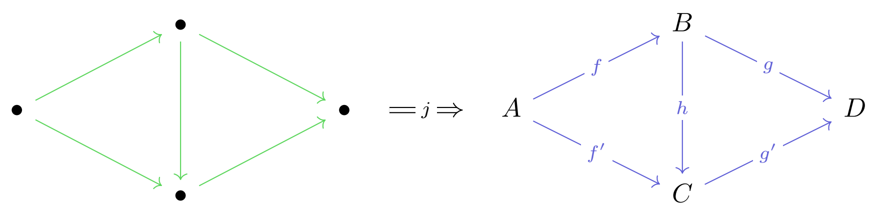
Example 2. *Here are some examples of functors:*

- *For any category C, the identity functor $id_{\mathsf{C}} : \mathsf{C} \to \mathsf{C}$ assigns each object and morphism to itself. These functors are the identity morphisms in Cat.*
- *For any two categories C, D and some object $X \in \mathsf{D}$ the constant functor $c_X : \mathsf{C} \to \mathsf{D}$ assigns every object in C to X and every morphism in C to id_X.*
- *There exists a functor $U : \mathsf{Grp} \to \mathsf{Set}$ that assigns every group its underlying set and every group homomorphism its underlying function, forgetting about the group law preserving property.*
- *Conversely, there exists a functor $F : \mathsf{Set} \to \mathsf{Grp}$ that assigns every set the free group generated by the set, and every function its natural free extention. These two functors form an example of an adjunction.*
- *Given two partially ordered sets, functors between their induced categories are precisely the order preserving maps.*
- *The functor $\pi_1 : \mathsf{Top}_\bullet \to \mathsf{Grp}$ sends a (path-connected) pointed topological space to its first fundamental group.*
- *For a category C and an object $X \in \mathsf{Ob}(\mathsf{C})$, the functor $\mathsf{C}(X, -) : \mathsf{C} \to \mathsf{Set}$ sends $Y \in \mathsf{Ob}(\mathsf{C})$ to $\mathsf{C}(X, Y)$ and morphisms $f \in \mathsf{C}(Y, Z)$ to pushforwards $\mathsf{C}(X, f) = f_* : \mathsf{C}(X, Y) \to \mathsf{C}(X, Z)$ under the rule $g \mapsto f \circ g$.*
- *Analogously, $\mathsf{C}(-, X) : \mathsf{C} \to \mathsf{Set}$ is also a functor, sending morphisms $f \in \mathsf{C}(Y, Z)$ to their pullback $f^* : g \mapsto g \circ f$.*
- *Given the category Man of smooth manifolds, the functor $T : \mathsf{Man} \to \mathsf{Man}$ turns each manifold M into its tangent bundle TM, and a smooth map f into the differential df.*

Functors provide a neat and elegant way to talk about (commutative) diagrams formally.

Definition 4. *A diagram of shape J over C is a functor $j : \mathsf{J} \to \mathsf{C}$.*

The shape J is to be seen as a "sketch" or a "blueprint" on how the diagram looks, whilst the functor j tells which objects and morphisms in C are present in the diagram. In the case of the diagram previously given, it would look a bit like this:

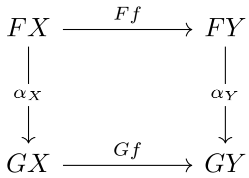


The category J is depicted in green, and C in blue.

1.2 Natural Transformations

Given two categories C, D we may consider the collection of all functors $F : \mathsf{C} \to \mathsf{D}$, denoted $[\mathsf{C}, \mathsf{D}]$ as a shorthand for $\mathsf{Cat}(\mathsf{C}, \mathsf{D})$. Turning this set into a category would be nice, but for the morphisms to be sensible we need to be able to transform functors in a natural way.

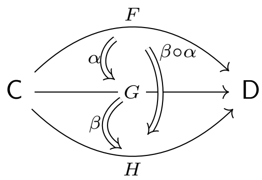
Definition 5. *Let $F, G : \mathsf{C} \to \mathsf{D}$ be functors. A* ***natural transformation*** *between two functors $\alpha : F \implies G$ consists of à collection of morphisms $\alpha_X : FX \to GX \in \mathsf{D}(FX, GX)$, every morphism associated to a single object in C, such that the following diagram commutes for every choice of $X, Y \in \mathsf{C}$:*



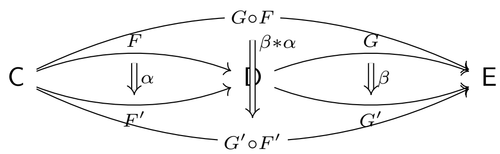
The collection of maps α is said to satisy the naturality condition imposed by the above diagram.

Example 3. *Natural transformations are a bit more sophisticated and as an effect the examples will be rather abstract:*

- *Given a functor $F : \mathsf{C} \to \mathsf{D}$, the identity natural transformation $id_F : F \implies F$ has identity morphisms id_{FX} as components.*
- *The double dual functor $(-)^{**} : \mathsf{Vect}_K \to \mathsf{Vect}_K$ makes for an elegant natural transformation $J : id_{\mathsf{Vect}_K} \implies (-)^{**}$. The components of this transformation are the natural injections $j : V \to V^{**}$ sending a vector v to the functional $w \mapsto w(v)$ for any given linear functional $w \in V^*$. There are two noteworthy things about this, namely that there does not exist an analogous natural transformation for the first dual, considering the natural maps would depend on a choice of basis; and secondly, when restricting J to finite dimensional vector spaces, it becomes a* ***natural isomorphism****.*
- *There are two functors $\pi_n, H_n : \mathsf{Top}_\bullet \to \mathsf{Grp}$, the first one maps a pointed topological space to its nth fundamental group whereas the second one maps to the nth homology class. We can define a natural transformation $h_n : \pi_n \implies H_n$ whose components are called Hurewicz homomorphisms, mapping homotopy classes of paths to generators in the homology class.*
- *Given three functors $F, G, H : \mathsf{C} \to \mathsf{D}$ and two natural transformations $\alpha : F \implies G, \beta : G \implies H$, we can define their* ***vertical composition*** *$\beta \circ \alpha : F \implies H$ as the natural transformation with components $(\beta \circ \alpha)_X = \beta_X \circ \alpha_X$ for all $X \in \mathsf{Ob}(\mathsf{C})$.*



- *Given four functors $F, F' : \mathsf{C} \to \mathsf{D}; G, G' : \mathsf{D} \to \mathsf{E}$, and two natural transformations $\alpha : F \implies F', \beta : G \implies G'$, we can define their* ***horizontal composition*** *$\beta * \alpha : G \circ F \implies G' \circ F'$ as the natural transformation with components $(\beta * \alpha)_X = \beta_{F'X} \circ G(\alpha_X) = G'(\alpha_X) \circ \beta_{FX}$*



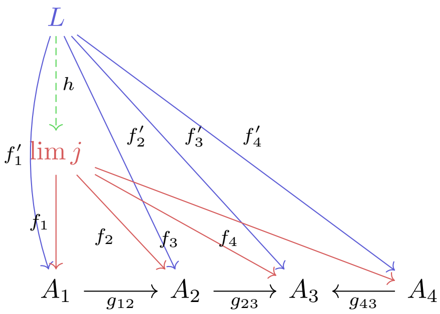
1.3 Limits

A trend in modern mathematics is that one is not as much interested in the definition of an object, but rather what properties an object has. Category theory is no different; many of its elementary concepts like products and equalizers are defined in terms of *universal properties*, encoded in a suitable limit diagram.

Definition 6. *Let $j : \mathsf{J} \to \mathsf{C}$ be a (not necessarily commuting) diagram in C with objects $(A_i)_{i \in I}$. The limit of the diagram j, if it exists, satisfies the following:*

- *There exists an object $\lim j \in \mathsf{Ob}(\mathsf{C})$ and a collection of morphisms $(f_i : \lim j \to A_i)_{i \in I}$ such that for each morphism $g_{st} : A_s \to A_t$, $g_{st} \circ f_s = f_t$*
- *If there exists another object L and a collection of morphisms $(f'_i : L \to A_i)_{i \in I}$ satisfying the above condition, then there exists a unique morphism $h : L \to \lim j$ such that $f_i \circ h = f'_i$ for all $i \in I$.*

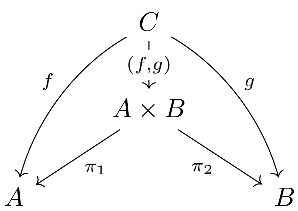
We may see the objects L that satisfy the first condition as *cones* over j, where the diagram is drawn on a flat plane and the object L with its morphisms are drawn as the tip and the legs of a cone respectively. Note that we only require the legs to commute. The limit itself is also a cone, except that in a way it is the best fit, every cone factors through the limit by the morphism h. An example of a limit cone is drawn below, the diagram j is given in black, the limit cone is drawn in red, another cone is drawn in blue and the unique mediating morphism is given in green.



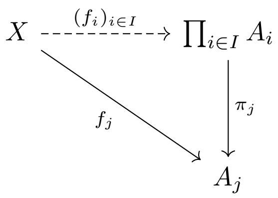
Example 4. • *Limits of the empty diagram are also called terminal objects. Although every object in a category fits the first criterion in definition 6, a terminal object T is characterized by that for every object X in the category there exists a unique terminal map $\tau : X \to T$. Terminal objects in the category of sets are singleton sets. Note that in a category a terminal object may not exist, for example in the category Field_p of fields with characteristic p.*

- *The limit of an one-object diagram of A is paradoxically more trivial, as the limit is given by (any object isomorphic to) the object A itself, together with the identity map (or any isomorphism). For any object L and morphism $f : L \to A$, the mediating morphism h is chosen to be $h = f$, as $f \circ id_A = f$.*

- *The limit of a two-object diagram with objects A, B and no nontrivial morphisms, if it exists, is denoted $A \times B$. For any object C and morphisms $f : C \to A, g : C \to B$ the mediating morphism is denoted $h = (f, g)$.*

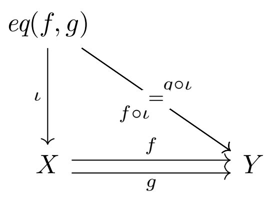


- *The previous three are examples of a more general family of limits called products, which are particularly useful to create ordered tuples in the context of sets. In general, a product may be defined as a limit of an indexed collection of objects $(A_i)_{i \in I}$. The morphisms provided by the limit are called canonical projections, and the mediating morphism is called a product morphism.*

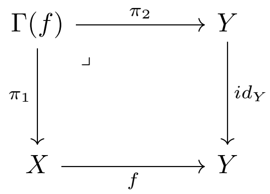


Definition 7. *The diagrams and cones associated to products are called* ***product diagrams*** *and* ***product cones*** *respectively.*

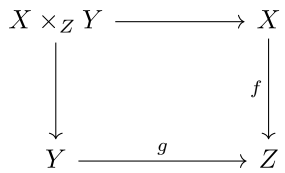
- *Given a diagram consisting of two objects X, Y and two parallel morphisms $f, g : X \to Y$, the limit is called the equalizer of f and g, denoted $eq(f, g)$. In the context of sets, and many algebraic structures with a base set, this will be a subset on which f, g agree, that is, $eq(f, g) = \{x \in X : f(x) = g(x)\}$.*



- *Given a diagram with three objects* X, Y, Y *and morphisms* $f : X \to Y$ *and* $id_Y : Y \to Y$*, the limit of this diagram is called the graph of* f*, denoted* $\Gamma(f)$*. Again, in the context of sets this object can be written out, as* $\Gamma(f) = \{(x, y) \in X \times Y : f(x) = y\}$*.*



- *The previous two examples are specific instances of pullbacks, which are limits over diagrams of the shape* $\bullet \to \bullet \leftarrow \bullet$*. For sets, it picks out pairs of elements where two functions agree, in the diagram below one has* $P = \{(x, y) \in X \times Y : f(x) = g(y)\}$*.*



Note that when Z *in the above diagram is a terminal object, then the pullback diagram corresponds to a binary product.*

Remark 1. *We use the notation* (f, g) *and* $(f_i)_{i \in I}$ *to denote binary and arbitrary product morphisms respectively. Note that in the definition given above, the morphisms all have the same domain. It is sometimes more convenient to use the notation* $(f, g) : X_1 \times X_2 \to Y_1 \times Y_2$ *given* $f : X_1 \to Y_1$*,* $g : X_2 \to Y_2$ *instead. In this article we will use both notations, as the domains of the morphisms will remove any ambiguity.*

2 Finite Lawvere Theory

2.1 Motivation

A key topic in algebra is the study of algebraic structures, like groups and rings, and the structure preserving maps between them, like group and ring homomorphisms. One may note that the definitions of these structures are usually not dissimilar, defining a structure to be a base set endowed with operations and elements that satisfy a set of constraints. The definitions of the corresponding homomorphisms then look even more identical: maps that respect the given operations and restrictions. In this section we seek to provide a framework to generalise the notion of an algebraic structure.
To do this, we may look first at a familiar structure to draw inspiration from its traditional introduction in mathematics:

Definition 8. *A group $(G, \cdot, e)$ is a triple consisting of*

- *A set G*
- *A binary operation $\cdot : G \times G \to G$ called the group law*
- *An element $e \in G$ called the identity or neutral element*

satisfying the following properties

- *For all $g \in G$, one has $g \cdot e = e \cdot g = g$*
- *For all $g \in G$, there exists an inverse denoted g^{-1} such that $g \cdot g^{-1} = g^{-1} \cdot g = e$*
- *For all $x, y, z \in G$, one has $(x \cdot y) \cdot z = x \cdot (y \cdot z)$*

A group homomorphism is a map $\varphi : (G, \cdot, e_G) \to (H, *, e_H)$ such that for all $g, h \in G$ one has $\varphi(g \cdot h) = \varphi(g) * \varphi(h)$

Some textbooks may omit the identity element from the triple and introduce it in the constraints bound by an existential quantifier instead.
This definition highlights the split between the data and the restrictions of the structure. It is therefore natural to try to mimic or accomodate for this pattern in our supposed definition of an algebraic structure:

Attempt 1. *An algebraic structure is a set together with a collection of operations and restrictions on these operations*

This definition, of course, is way too crude. First we have to agree on the notion of an operator, and secondly and more importantly we have to decide what a restriction on these operators entails.
If we look at conventional operators like multiplication and addition, we may agree that they take two elements out of a set and assign it to a new element. Whilst defining an operator on a set X to be a map $o : X \times X \to X$ is a good starting point, it is quite restrictive. Rather, it is better to consider tuples of arbitrary size instead. An operation on X then becomes a map $o : X^n \to X$. The size n of the tuple determines the *arity* of the operation, which in turn is called *n-ary*.
The restrictions are arguably more problematic. Mainly because they are statements about operators rather than mathematical objects, so without proper justification we cannot stow them away in a designated set. Let us take another look again at the restrictions imposed on a group, but now rehashed into the language of formal logic:

- $\forall_{g \in G} : g \cdot e = g \wedge e \cdot g = g$
- $\forall_{g \in G} \exists_{h \in G} : g \cdot h = e \wedge h \cdot g = e$
- $\forall_{x \in G} \forall_{y \in G} \forall_{z \in G} : (x \cdot y) \cdot z = x \cdot (y \cdot z)$

Of course, one may be satisfied with this and define an algebraic structure to be a triple consisting of a base set of elements, a set of operators and a collection of logical statements that these operators abide, but one must agree that the above statements do not look uniform and similar. For starters, the first condition can be split up in two parts, but the second cannot because of the existential quantifier.
We can propose one immediate improvement: by using equations in functions we eliminate the need of the clumsy universal quantifiers. After all, given $f, g : X \to Y$,

$$f = g \iff \forall_{x \in X} : f(x) = g(x)$$

And similarly and more importantly, for $f, g : X^n \to X$,

$$f = g \iff \forall_{x_1 \in X} \dots \forall_{x_n \in X} : f(x_1, \dots, x_n) = g(x_1, \dots, x_n)$$

If we now set $a_r(x, y, z) = x \cdot (y \cdot z)$ and $a_l(x, y, z) = (x \cdot y) \cdot z$, the last group criterion can be shortened to the simple $a_l = a_r$.
Of course, it would be even better if we could write out this equality in terms of the multiplication instead. Using the definition of a product morphism, we see that $a_r = \cdot \circ (id, \cdot)$, where $(id, \cdot) : G^3 = G \times G^2 \to G \times G = G^2$ is given by $(id, \cdot)(x, y, z) = (x, y \cdot z)$. Functions can also be used to rid of the existential quantifiers, which is apparant after inspecting the very definition of a function:

$$f : X \to Y \text{ is a function } \iff \forall_{x \in X} \exists!_{y \in Y} : y = f(x)$$

The caveat here is that the function assigns every element to a *unique* element in the codomain, but since inverses in groups are unique, this idea can still be applied. If we consider the single point set $\{*\}$ as the domain, and the group as the codomain, functions become entirely determined by what they map the single element in the domain to, and therefore can be used to encode constants. In the context of groups, this means we can replace the identity element e with an "identity element pointer function" $e : \{*\} \to G$. Let $i : G \to G$ denote the map $i(x) = x^{-1}$, let $\Delta : G \to G^2$ denote the diagonal map $\Delta(x) = (x, x)$ and $\epsilon : G \to G$ denote the constant map $\epsilon(x) = e$. The second group criterion now becomes

$$\begin{aligned} \cdot \circ (id, i) \circ \Delta &= \epsilon \\ \cdot \circ (i, id) \circ \Delta &= \epsilon \end{aligned}$$

More verbosely, let us see how both sides of the first equation act on an arbitrary element $x \in G$:

$$\begin{aligned} (\cdot \circ (id, i) \circ \Delta)(x) &= (\cdot \circ (id, i))(x, x) \\ &= (\cdot)(id(x), i(x)) \\ &= (\cdot)(x, x^{-1}) \\ &= x \cdot x^{-1} \\ \epsilon(x) &= e \end{aligned}$$

Thus, $x \cdot x^{-1} = e$ for all $x \in G$. A similar computation can be made for the second statement.
We can now employ these insights to make a second, more informed, attempt of defining an algebraic structure. To avoid having to put formal logical statements into sets, we can use pairs of functions instead.

Attempt 2. *An algebraic structure is a triple $(X, \mathcal{O}, \Lambda)$ where X is a set, $\mathcal{O}$ is a collection of operations $o : X^n \to X$ and Λ is a collections of pairs of functions (f, g) of the same arity such that $f(x_1, \dots, x_n) = g(x_1, \dots, x_n)$ for all $x_i \in X$.*

Still, this might not be the best of ideas. We are placing functions in Λ and forcing them to be equal.
Rather, we are interested in the **expressions** formed by the operations in $\mathcal{O}$, which can be formalized as elements of a free operad[Lei04] on $\mathcal{O}$.
As to not open that metaphorical can of worms, we will quickly and inductively define a set E of n-ary expressions:

- for every n-ary operation $o \in \mathcal{O}$, E contains an n-ary expression e_o
- for every n-ary expression e and $k_1, k_2, \dots, k_n$-ary expressions $e_1, e_2, \dots, e_n$, a $k_1 + k_2 + \dots + k_n$-ary expression $e \circ (e_1, e_2, \dots, e_n)$

Furthermore, given an algebraic structure $(X, \mathcal{O})$, there exists a way of evaluating those expressions ev_X such that

$$ev_X(e_o)(x_1, \dots, x_n) = o(x_1, \dots, x_n) \tag{1}$$

$$\begin{aligned} &ev_X(e \circ (e_1, \dots, e_n))(x_1, \dots, x_{k_1 + \dots + k_n}) = \\ &ev_X(e)(ev_X(e_1)(x_1, \dots, x_{k_1}), ev_X(e_2)(x_{k_1+1}, \dots, x_{k_1+k_2}), \dots, ev_X(e_n)(x_{k_1+\dots+k_{n-1}+1}, \dots, x_{k_1+\dots+k_n})) \end{aligned} \tag{2}$$

Λ then is a collection of pairs of expressions that evaluate to the same function under ev_X. For the sake of readability and in order to avoid cumbersome notation, we will use expressions and functions interchangeably. For most[4] intents and purposes this attempt suffices in providing a way of understanding how a structure

[4]Following this line of definitions, one may encounter difficulties defining fields, something we will see in section 3.

works internally, but the study of homomorphisms of similar structures is often if not always more insightful. Intuitively, we want a homomorphism to respect the operations present in $\mathcal{O}$, like how a group homomorphism preserves the group law. Therefore, if we want to introduce the notion of a structure preserving map, we not only need a function on the base sets, but also a correspondence in the set of operations.

Attempt 3. *Let $(X, \mathcal{O}, \Lambda), (Y, \mathcal{U}, \Xi)$ be algebraic structures. A homomorphism φ consists of two functions $\varphi_0 : X \to Y$ and $\varphi_1 : \mathcal{O} \to \mathcal{U}$ such that for every n-ary operation $o \in \mathcal{O}$ the following holds:*

$$\varphi_0(o(x_1, \dots, x_n)) = \varphi_1(o)(\varphi_0(x_1), \dots, \varphi_0(x_n))$$

There are two questions lurking behind this definition. The first one is inquiring about the involvement of Λ and Ξ in this definition. After all, latching back on our ever trustful example of groups, a group homomorphism only makes sense between groups. It turns out that this is of no concern, if we restrict our codomain to the image of the homomorphism, and work under the assumption that Λ only contains pairs of operations or products of operations. This however is a bit unsatisfying to write out.
The second question that may arise is whether φ_1 should be a bijection. With the following two examples we would like to argue that this is not inheritally necessary.

Example 5. *Let $\mathcal{O} = \varnothing = \Lambda$ and $\mathcal{U} \neq \varnothing$. An algebraic structure $(X, \mathcal{O}, \Lambda)$ models a basic set, and a homomorphism $\varphi : (X, \mathcal{O}, \Lambda) \to (Y, \mathcal{U}, \Xi)$ is a regular function. Even though φ_1 is not surjective, one may argue that all the structure on X is preserved.*

Example 6. *Let $\mathcal{O} = \{p_1, p_2\}$ contain two nullary functions, thus corresponding to two basepoints, and similarly let $\mathcal{U} = \{q_1\}$ be a set with one nullary function. The structure $(Y, \mathcal{U}, \varnothing)$ is called a pointed set, and maps between pointed sets are expected to preserve the basepoint. Similarly, $(X, \mathcal{O}, \varnothing)$ is a doubly pointed set. A homomorphism $\varphi : (X, \mathcal{O}, \varnothing) \to (Y, \mathcal{U}, \varnothing)$ necessarily has $\varphi_1(p_1) = \varphi_1(p_2) = q_1$, and thus is not surjective. This homomorphism maps both basepoints in X to the one basepoint in Y.*

Still, by far the most use we get out of homomorphisms is when the two structures are "similar enough". We often want $\mathcal{O}, \mathcal{U}$ and Λ, Ξ to be equal. This motivates us to decouple the intrinsic structure of the theory, encoded by $(\mathcal{O}, \Lambda)$, from the base set X and the operations acting on it. As an added bonus, we can encode the class of operations in $\mathcal{O}$ without assigning explicit symbols, meaning e.g. for groups the multiplicative notation is not fixed.

Definition 9. *An algebraic theory is a pair $(\mathcal{O}, \Lambda)$, where $\mathcal{O} = \bigcup_{i=0}^{n} \mathcal{O}_i$ is a collection of operations with a given arity, and $\Lambda \subset E \times E$ is a collection of pairs of expressions in $\mathcal{O}$.*
An algebraic structure of type $(\mathcal{O}, \Lambda)$ is a pair $(X, ev_X : \mathcal{O} \to O)$ where X is a set, and ev_X is a map that assigns every m-ary operation o to a function $ev_X(o) : X^m \to X$.

Remark 2. *It should be noted that using the properties of E as a free operad over $\mathcal{O}$, $ev_X : \mathcal{O} \to O$ can be extended to $ev_X : E \to O$ using equations (1) and (2). Naturally, this extention respects the arity of the expressions in E.*
For the remainder of the thesis we will be using ev_X as defined over its extended domain.

The definition of a homomorphism also adapts quite nicely:

Definition 10. *Let $(X, ev_X), (Y, ev_Y)$ be two algebraic structures of type $(\mathcal{O}, \Lambda)$. A homomorphism $\varphi : X \to Y$ is a map such that*

$$\varphi(ev_X(o)(x_1, \dots, x_m)) = ev_Y(o)(\varphi(x_1), \dots, \varphi(x_m))$$

for every m-ary operation o in $\mathcal{O}$.

Definition 11. *As a reference frame, we define the category $\mathsf{A}_{\mathcal{O},\Lambda}$ whose objects are algebraic structures as defined in definition 9, and whose morphisms are homomorphisms as defined in definition 10.*

2.2 Finite Lawvere Templates

2.2.1 Product Preserving Functors

The use of commutative diagrams to impose restrictions on our algebras is a further hint to how useful category theory promises to be in this setting. Considering that all algebraic structures are related to a base set of sorts, we may want to relate them to special functors $F : \mathsf{C} \to \mathsf{Set}$ from a template category C.

Definition 12. *A **product preserving functor** between categories C, D is a functor $F : \mathsf{C} \to \mathsf{D}$ such that for every product cone $j : \mathsf{J} \to \mathsf{C}$ the cone $F \circ j$ is a product cone in D.*

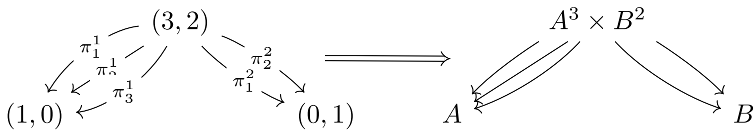


Figure 1: An example of the projection morphisms given in the case $k = 2, n_1 = 2, n_2 = 3$. A comparison is made to canonical projections of products of sets.

As an effect, this means that every collection of objects $(A_i)_{i \in I} \in \mathsf{C}$ for which the product $\prod_{i \in I} A_i$ exists in C, the product $\prod_{i \in I} FA_i$ exists in D and $F\left(\prod_{i \in I} A_i\right) = \prod_{i \in I} FA_i$. Moreover, tuples of functions are mapped in a intuitive manner, for example $F((f, g)) = (F(f), F(g))$. Product preserving functors are part of a full subcategory of the functor category $[\mathsf{C}, \mathsf{D}]$, which we denote as $[\mathsf{C}, \mathsf{D}]_\Pi$.
With product preserving functions giving us a way to interpret the actions of n-ary operations, the original question can be reformulated into the following: *Given a category* A *of algebraic structures and homomorphisms, is there a category* a *such that* $[\mathsf{a}, \mathsf{Set}]_\Pi \equiv \mathsf{A}$*?*

2.2.2 Templates

In the template category a mentioned in the definition of the previous section, the objects will often be products of some generating objects with themselves, with some morphisms representing the n-ary operations in between. The product preserving functors are then completely determined by where it maps the generating objects in question to, and how it forms the n-ary operations. This behavior will be outlined in section 3.

Definition 13. *Let* $n_1, \ldots, n_k$ *be natural numbers and* $[n_i] = \{0, \ldots, n_i - 1\}$. *A category* T *is a* ***(finite lawvere) template category*** *if it satisfies the following:*

- $\mathsf{Ob}(\mathsf{T}) = [n_1] \times \cdots \times [n_k]$
- $(a_1, \ldots, a_k) \times (b_1, \ldots, b_k) = (a_1 + b_1, \ldots, a_k + b_k)$ *whenever* $a_i + b_i < n_i$ *for all* $0 < i \leq k$

Definition 14. *A category* A *for which* $\mathsf{A} \equiv [\mathsf{T}, \mathsf{Set}]_\Pi$ *for some template category* T *is called a* ***finite lawvere algebra***

We say that T is a template category on k generators or with k generating objects.
Since the objects of the template category are finite, using an inductive argument the following equivalent generalisation of the second condition can be stated[5]:

Theorem 2.1. $(a_1, \ldots, a_k) = (1, \ldots, 0)^{a_1} \times \cdots \times (0, \ldots, 1)^{a_k}$

Proof. The forward direction relies on induction where the base case is clear and the inductive step is given by $(c_1 + d_1, \ldots, c_k + d_k)$ where c_i is carried over from the previous iteration, and $d_i = \begin{cases} 1 & c_i < a_i \\ 0 & c_i = a_i \end{cases}$. The backwards direction is given by breaking up the product and rewriting. □

Corollary 1. $\prod_{j \in I}(a_{1,j}, \ldots, a_{k,j}) = (\sum_{j \in I} a_{1,j}, \ldots, \sum_{j \in I} a_{k,j})$ *whenever* $\sum_{j \in I} a_{i,j} < n_j$ *for all* $0 < i \leq k$.

The generality of the definition of a template category makes it so we can model structures that have more than one base set, like an R-module M which has two different sets associated with it that interact with each other.

It may be worthwhile to write out the case $k = 1$ to form a better intuition around the definition.

Definition 15. *A category* T *is a* ***template category with 1 generating object*** *if it satisfies the following:*

- $\mathsf{Ob}(\mathsf{T}) = [n] = \{0, \ldots, n - 1\}$
- $a \times b = a + b$ *whenever* $a + b < n$. *Equivalently,* $i = \prod_i 1$

Theorem 2.1 tells us that an object $i \in \mathsf{Ob}(\mathsf{T})$ is the i-fold product of 1 with itself. This has the added benefit that product preserving functors $F : \mathsf{T} \to \mathsf{Set}$ are on objects uniquely defined by the set $F(1)$ as $F(i) = F(1)^i$. Naturally this property holds for general template categories:

[5] Technically, the statements are equivalent up to isomorphism of objects, as nowhere it is given that $X \times Y = Y \times X$. However, for the sake of clarity this detail is ignored.

Remark 3. *If it is the case that* T *has a simple structure where every morphism can be derived as a finite composition of "base morphisms" and product morphisms*[6]*, then a product preserving functor* F *is completely determined by the following data:*

- *The* k *sets* $F((0,\dots,1,\dots,0))$
- *The collection of functions* $\{F(\alpha)\}$*, where the* α*'s are the aforementioned "base morphisms"*

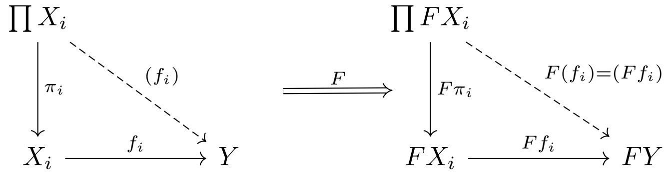
In our discussion on algebraic structures, these "base morphisms" make up the collection of operations.

Proof. Let $(a_1,\dots,a_k)$ be an object of T. Then, by theorem 2.1

$$F((a_1,\dots,a_k)) = F((1,\dots,0)^{a_1}\times\cdots\times(0,\dots,1)^{a_k}) = (F((1,\dots,0)))^{a_1}\times\cdots\times(F((0,\dots,1)))^{a_k}$$

Which is a product consisting only of sets of the form $F((0,\dots,1\dots,0))$.
Definition 13 allows for four types of morphisms: projection, primitive, product and composite. Since they are obtained by finite applications of the last two rules, we are able to retrace them to projection and primitive morphisms. Product preserving functors preserve how these morphisms interact with each other:

- *projections are mapped to canonical projections of product sets*: Given the product cone of $(a_1,\dots,a_k)$ over its components $(0,\dots,1,\dots,0)$, the projections π^i_j are mapped to the corresponding canonical projections of the product in the category Set.
- *images of "base morphisms" depend on* $\{F(\alpha)\}$ trivially.
- *product morphisms depend only on their components*: product morphisms $(f^i)_j$ arise as the universal morphism of a suitable product cone whose legs are given by f^i_j. By product preservingness, $F((f^i)_j) = (F(f^i))_j$.
- *strings of formal compositions depend only on their parts*: this follows from the property of functors, $F(f\circ g) = F(f)\circ F(g)$.



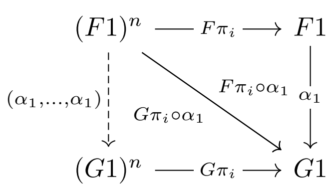
□

2.3 Homomorphisms

Once we find the algebraic structures to be functors $F:\mathsf{T}\to\mathsf{Set}$, it is natural to consider the natural transformations between these functors, and compare them to how conventional algebraic homomorphisms behave.

Definition 16. *Let* $\mathsf{A}_\mathsf{T} = [\mathsf{T},\mathsf{Set}]_\prod$ *be a finite lawvere algebra arising from a template* T*. A homomorphism in* A_T *between two objects* $F,G:\mathsf{T}\to\mathsf{Set}$ *is defined to be a natural transformation* $F \implies G$

It should come as no surprise that a category modelling an algebraic theory should have homomorphisms as its morphisms.
A first thing to notice is how the components of the natural transformation α depend only on the components of $\alpha_{(0,\dots,1,\dots,0)}$. For simplicity, only the case for template categories with one generating object will be portrayed. The product preserving functors F,G map projections to canonical projections on sets. As such, a collection of n morphisms $F\pi_i\circ\alpha_1$ form. By universal property of $(G1)^n$, there exists a unique morphism $h:(F1)^n\to(G1)^n$, such that $G\pi_i\circ h = \alpha_1\circ F\pi_i$. Note that the product morphism $(\alpha_1,\dots,\alpha_1)$ satisfies this property, thus by uniqueness $h = (\alpha_1,\dots,\alpha_1)$.



[6] More commonly known as a free construction, which we will touch upon later

Now compare this to the naturality condition of the natural transformation applied to the projections π_i:

$$\begin{array}{ccc} Fn & \xrightarrow{F\pi_i} & F1 \\ \Big\downarrow{\scriptstyle \alpha_n} & & \Big\downarrow{\scriptstyle \alpha_1} \\ Gn & \xrightarrow{G\pi_i} & G1 \end{array}$$

Since $(F1)^n = Fn$ and $(G1)^n = Gn$, we conclude $\alpha_n = (\alpha_1, \ldots, \alpha_1)$.
As mentioned, this result cleanly carries over to natural transformations between functors from template categories with more than one generator.
The natural transformations have the nice property that they preserve the actions of the morphisms in the template category in a certain way. For example, compare the following diagram describing the property of a group homomorphism h preserving the group law from the groups $(G, \cdot), (G', *)$

$$\begin{array}{ccc} G \times G & \xrightarrow{\quad \cdot \quad} & G \\ \Big\downarrow{\scriptstyle (h,h)} & & \Big\downarrow{\scriptstyle h} \\ G' \times G' & \xrightarrow{\quad * \quad} & G' \end{array}$$

That is, for all $x, y \in G$ one has $h(x \cdot y) = h(x) * h(y)$, to the diagram portraying the naturality square for an arbitrary morphism from the template category:

$$\begin{array}{ccc} Fn & \xrightarrow{Ff} & F1 \\ \Big\downarrow{\scriptstyle (\alpha_1,\ldots,\alpha_1)} & & \Big\downarrow{\scriptstyle \alpha_1} \\ Gn & \xrightarrow{Gf} & G1 \end{array}$$

That is, for all $x_1, \ldots, x_n \in F1$ one has $\alpha_1(Ff(x_1, \ldots, x_n)) = Gf(\alpha_1(x_1), \ldots, \alpha_1(x_n))$.

2.4 Morphisms between Template Categories

Given that template categories are categories on their own, it is natural to consider a subcategory of Cat containing only template categories. As per usual, we would like the morphisms of this subcategory to preserve the structure of the template categories.

Definition 17. *We denote the category of (finite lawvere) template categories and product preserving functors between them with* FLTCat.
Furthermore, we denote the category of finite lawvere algebras with Alg*, whose objects are finite Lawvere algebras* $[\mathsf{T}, \mathsf{Set}]$ *and whose morphisms are product preserving functors between those algebras.*

Note that there is a distinct difference between the category of template categories FLTCat, and the category of product preserving functors from a fixed template into sets, denoted $[\mathsf{T}, \mathsf{Set}]_\Pi$. However, they are intertwined in a certain way: Given a morphism between template categories $F : \mathsf{T}_1 \to \mathsf{T}_2$, define the pullback $F^* : [\mathsf{T}_2, \mathsf{Set}]_\Pi \to [\mathsf{T}_1, \mathsf{Set}]_\Pi$ as $F^*G = G \circ F$.

$$\begin{array}{ccc} \mathsf{T}_1 & \xrightarrow{\qquad F \qquad} & \mathsf{T}_2 \\ & \Big\Downarrow{\scriptstyle \text{pullback}} & \\ \mathsf{Alg}_{\mathsf{T}_1} = [\mathsf{T}_1, \mathsf{Set}]_\Pi & \xleftarrow[F^*]{} & [\mathsf{T}_2, \mathsf{Set}]_\Pi = \mathsf{Alg}_{\mathsf{T}_2} \end{array}$$

Note that this relation is contravariant. Luckily, the induced pullback has a clear intuition.

Example 7. *Consider the canonical inclusion functor from the template category for magmas into the template category for groups. The pullback along this functor between the representations correspond to a forgetful functor from groups to magmas.*
As such, if the preimage of a morphism is empty, the pullback corresponds to "forgetting" about the operation in the representations.

Example 8. *Consider the surjective functor from the template category for doubly pointed sets to the template category for pointed sets. The pullback between representations corresponds to a functor assign to a pointed set the doubly pointed set with the same basepoint twice.*
As such, if the preimage of a morphism is a nonempty set, the pullback corresponds to "duplicating" the operation in the representations.

Example 9. *Consider the template category for monoids and the template category for rings. In this case, there are two natural choices for a functor between the templates, one sending the monoid law and identity to the addition and zero of the ring, and the other sending them to the multiplication and unit.*
This exhibits two things: the pullback functor is often neither injective nor surjective.
The ring of square matrices of a given size with the matrix product and the hadamard (entrywise) product both have the same additive group, and thus the pullback is not injective. Furthermore, every additive group is an abelian group, and there exists many monoids that are neither groups nor abelian.

This pullback relation is formalized by means of a bifunctor[7] $\Phi : \mathsf{FLTCat}^{\mathrm{op}} \times \mathsf{FLTCat} \to \mathrm{Hom}(\mathsf{Alg}, \mathsf{Alg})$, given by $\Phi(\mathsf{T}, \mathsf{S}) = [\mathsf{A}_\mathsf{S}, \mathsf{A}_\mathsf{T}]_\Pi$. Fixing templates T and S, this relation is not surjective. A counterexample is given by the (product preserving) restriction of a ring to its unit group; there is no suitable embedding for the template for groups into the template rings as the multiplicative inverse map does not exist.

2.5 Construction of template categories

We aim to characterise a template category by its bare ingredients, those being the base sets, operations and restrictions on those operations. In order to create a template category, we will try to leverage free constructions. A neocategory is a generalisation of a category where composition is replaced by a partial composition law, and therefore is a good candidate for the adjunction as it saves the burden of writing out all compositions. A natural first guess is to state the adjunction will factor through Cat (pictured below), assuming the free functor $F : \mathsf{Cat} \to \mathsf{TCat}$ exists:

$$\begin{array}{c} \mathsf{NeoCat} \\ U \uparrow \dashv \downarrow F \\ \mathsf{Cat} \\ U \uparrow \dashv \downarrow F? \\ \mathsf{Template} \end{array}$$

Remark 4. *The free construction of a category from a given neocategory can be made by identifying the necessary morphisms in the free category from the underlying quiver. Please refer to chapter II.7 of [Lan78] for more details. Neocategories are more thouroughly defined and placed in the context of sketches in [BE72]. For a more rigorous discussion on the free construction of a category from a neocategory, refer to [SH00].*

However, this schema fails on two accounts:

- The neocategory does not encode a sufficient product structure, and thus we are not able to denote certain restrictions. For example, associativity in groups is given by $m \circ (id, m) = m \circ (m, id)$, an equality containing two product morphisms.
- The category-template category adjunction assigns a template category freely, thus the product structure is ambiguous. A category with objects $1, 2, \ldots, n$ encoding a n-ary operation is identical to a category with objects $(1, \ldots, 0), \ldots, (0, \ldots, 1)$ modeling a structure with n base sets after applying the free construction.

We will rather provide a step-by-step recipe to create a template category from a given algebraic structure.
If we wish to identify morphisms in a category, the following notion will be useful:

Definition 18. *Let C be a category and $\sim_{\mathsf{C}(X,Y)}$ be a collection of equivalence relation on the morphism sets of C such that the following holds:*

- *Given $f, f' : X \to Y$ and $g, g' : Y \to Z$, if $f \sim f'$ and $g \sim g'$, then $g \circ f \sim g' \circ f'$.*

Then, the quotient category $\mathsf{C}/\sim$ is defined in the following way:

- $\mathsf{Ob}(\mathsf{C}/\sim) = \mathsf{Ob}(C)$
- $(\mathsf{C}/\sim)(X, Y) = \mathsf{C}(X, Y)/\sim_{\mathsf{C}(X,Y)}$
- $[g] \circ [f] = [g \circ f]$

[7] A bifunctor exhibits functoriality in both components

Lemma 2.2. *The quotient category is a well-defined category.*

Proof. Given that $\sim_{\mathsf{C}(X,Y)}$ are equivalence relations on the morphism classes of C, the object class and morphism sets of the quotient are well-defined. We thus only have to show that identities and composition are well-defined. Let $[f] : X \to Y$, $[g] : Y \to Z$ be two morphisms in the quotient. Taking representatives $f \in [f], g \in [g]$, we see that $g \circ f \in [g \circ f]$ is defined. The fact that this definition is independent from the choice of representative follows directly from the condition on the equivalence relations: if $f' \in [f], g' \in [g]$ are arbitrary representatives, we have $g \circ f \sim g' \circ f'$ and thus $[g \circ f] = [g' \circ f']$. □

Remark 5. *The choice of composition ensures the natural map* $\mathsf{C} \to \mathsf{C}/\sim$ *is functorial.*

Step 1: Creating an Empty Template

From here on, we will try to find a template category modelling an algebraic structure of type $(\mathcal{O}, \Lambda)$. We will first ensure the template category has enough product objects to model the operations and the laws. Let us denote n to be the largest arity of an operation or law present in $\mathcal{O}$, Λ respectively, whichever is higher.
We define the empty template category T_0 as follows:

- $\mathsf{Ob}(\mathsf{T}_0) = [n+1] = \{0, 1, \dots, n\}$
- $\mathsf{T}_0(i,j) = \{\hat{\pi}_\alpha \mid \alpha : [j] \to [i]\}$

That is, morphisms from i to j are symbols $\hat{\pi}_\alpha$ where α is a function from a j-element set to a i-element set. The composition in this category is given by $\hat{\pi}_\beta \circ \hat{\pi}_\alpha = \hat{\pi}_{\alpha \circ \beta}$.
These morphisms will later act as a combination of projections and diagonal maps, however we will refer to them as projections. To justify this choice, the following definition will explain how these morphisms will act concretely.

Definition 19. *Let* $\alpha : [i] \to [j]$ *be a function, and fix a set* X*. Define* $\pi_\alpha : X^j \to X^i$ *as follows:*
$\pi_\alpha(x_1, x_2, \dots, x_j) = (x_{\alpha(1)}, x_{\alpha(2)}, \dots, x_{\alpha(i)})$

Example 10. *When* $\alpha : [1] \to [n]$ *is a constant function sending* 1 *to* i*,* π_α *is just the projection on the* i*th coordinate. We may write* π_i *instead of* π_α *in this case.*

Example 11. *The unique map* $\alpha : \varnothing \to [n]$ *corresponds to the unique projection from* X^n *to the single point set* $X^0 = \{\star\}$*. This map may be denoted* τ*, and is called the terminal map. Note that* 0 *is the terminal object in the empty template.*

Example 12. *The projection corresponding to the unique map* $\delta_k : [k] \to [1]$ *sends* x *to the tuple* $(x, \dots, x)$ *(*k *times). This map, known as a diagonal map, will be denoted* $\pi_{\delta_k} = \Delta^k$*.*

Example 13. *Let* $\sigma \in S_k$ *be a permutation.* π_σ *permutes the coordinates of elements in* X^k *according to* σ*.*

Lemma 2.3. *The assignment* $\alpha \mapsto \pi_\alpha$ *is contravariant: for any two functions* $\alpha : [j] \to [k]$*,* $\beta : [i] \to [j]$*, one has* $\pi_{\alpha \circ \beta} = \pi_\beta \circ \pi_\alpha$*.*

Proof. This follows from the definition of $\pi_{\alpha\circ\beta}$:

$$\begin{aligned}
\pi_{\alpha\circ\beta}(x_1, \dots, x_k) &= (x_{(\alpha\circ\beta)(1)}, \dots, x_{(\alpha\circ\beta)(i)}) \\
&= (x_{\alpha(\beta(1))}, \dots, x_{\alpha(\beta(i))}) \\
(\pi_\beta \circ \pi_\alpha)(x_1, \dots, x_k) &= \pi_\beta(x_{\alpha(1)}, \dots, x_{\alpha(j)}) \\
&= (x_{\alpha(\beta(1))}, \dots, x_{\alpha(\beta(i))})
\end{aligned}$$

□

We now make the following observation:

Lemma 2.4. *In* T_0*, whenever* $i + j \leq n$*, the categorical product of* i *and* j *is equal to* $i + j$

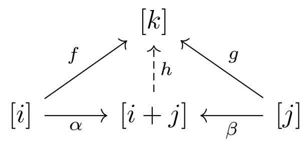
Proof. The definition of T_0 agrees with the opposite category corresponding to the partially ordered set $0 \leq 1 \leq 2 \leq \dots \leq n$, and in this poset category the coproduct is indeed given by a sum. Still, we will give the canonical projections and product morphisms ourself.
Let $i, j, k, \hat{\pi}_f : k \to i, \hat{\pi}_g : k \to j$ be arbitrary such that $i+j \leq n$. Furthermore, let $\alpha : [i] \to [i+j], \beta : [j] \to [i+j]$ be the injections into the first i and last j elements.

$$\alpha(x) = x$$
$$\beta(x) = x + i$$

Note that $0 \leq \alpha(x) < i$ and $i \leq \beta(y) < j$ for all $0 \leq x < i$ and $0 \leq y < j$ respectively.
Since all morphisms in T_0 are symbols corresponding to functions, we find two functions $f : [i] \to [k], g : [j] \to [k]$.
We define the morphism $h : [i + j] \to [k]$ as

$$h(x) = \begin{cases} f(x) & 0 \leq x < i \\ g(x - i) & i \leq x < j \end{cases}$$

We verify that the following diagram commutes:

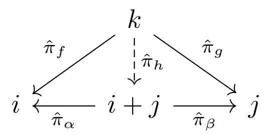


$$(h \circ \alpha)(x) = h(x) = f(x)$$
$$(h \circ \beta)(y) = h(y + i) = g(y)$$

And that h is unique to this property: if $h'(x) \neq h(x)$ for some $x \in [i + j]$, then either

$$x < i \implies h'(\alpha(x)) = h'(x) \neq f(x)$$
$$x \geq i \implies h'(\beta(x - i)) \neq h(\beta(x - i)) = g(x - i)$$

Replacing all the functions by their respective projections, and noting commutativity is preserved by the definition of composition in T_0, we obtain a product limit diagram.



□

Step 2: Creating the pretemplate

In the second step we will add the morphisms corresponding to the operations in our algebraic structure, whilst trying to preserve the product structure on T_0. We define the pretemplate category as a quotient on the following:

- $\mathsf{Ob}(\mathsf{T}_1) = \mathsf{Ob}(\mathsf{T}_0)$
- $\mathsf{T}_1(i, j) = \mathsf{T}_0(i, j)$ when $j \neq 1$, and
- $\mathsf{T}_1(i, 1) = \mathsf{T}_0(i, 1) \cup \Xi_i = \mathsf{T}_0(i, 1) \cup \{\xi_o \mid o \in \mathcal{O}_i\}$

where $\mathcal{O}_i \subset \mathcal{O}$ is the set of i-ary operations. $\Xi = \bigcup \Xi_i$ is a shallow copy of symbols of $\mathcal{O}$ for notational purposes.
In order to preserve the product structure, we add the following morphisms recursively:

- For any two morphisms $f : j \to k, g : i \to j$ their formal composition $f \diamond g$
- For any collection of morphisms $f_i : j \to k_i$ where $\sum_i k_i \leq n$, a formal product (f_i)

T_1 forms a category with $\diamond$ as its composition rule and the pre-existing identity morphisms.
Unfortunately, the pretemplate category T_1 does not have products like T_0 does. We will solve this problem in the next step.

Step 3: Shaping the Template

We define a collection $\sim_{\mathsf{T}_1(i,j)}$ of equivalence relations generated by the following:

- $h \sim f \diamond g$ for all $h = f \circ g$ in T_0
- $f_i \sim \hat{\pi}_\alpha \diamond (f_i)$ where $\hat{\pi}_\alpha$ is the projection onto the k_i coordinates in the $\sum_{j=1}^{i-1} k_j$th to $\sum_{j=1}^{i} k_j$th places

- $f \sim g$ for $(f,g) \in \Lambda$[8]

This accomplishes three things: the first condition ensures all compositions in T_0 are reflected. In particular, this brings back the intrinsic product structure between the projection moprhisms. The second condition ensures this product structure is bestowed upon the rest of T_1. Finally, the third condition ties our collection of wanted restrictions into it.
We then simply define the template category as the quotient

$$\mathsf{T} = \mathsf{T}_1/\sim$$

Lemma 2.5. T *is a template category with 1 generating object. Furthermore, the inclusion* $i : \mathsf{T}_0 \to \mathsf{T}$ *is functorial.*

Proof. Firstly, T is a category by lemma 2.2.
The template T satisfies by definition $\mathsf{Ob}(\mathsf{T}) = \mathsf{Ob}(\mathsf{T}_1) = \mathsf{Ob}(\mathsf{T}_0) = [n]$. For the first claim it rests to prove that $i \times j = i + j$ whenever $i + j \leq n$.
Fortunately, this also follows immediately from the definition of T. Let $f : k \to i, g : k \to j$ be arbitary morphisms. Because $i + j \leq n$, there is a morphism $(f,g) \in \mathsf{T}_1$ and as such a representative of (f,g) in T. Furthermore, the quotient rules that $f = \hat{\pi}_\alpha \circ (f,g), g = \hat{\pi}_\beta \circ (f,g)$ in T where α, β are the functions we described in step 1 to be the canonical projections.
To show $i : \mathsf{T}_0 \to \mathsf{T}$ is functorial, we need to show two things:

- identities are preserved. Recall that in both steps of the construction the identities are preserved.
- composition is preserved. Given a composition $f \circ g \in \mathsf{T}_0$, it suffices to show that the formal composition $i(f) \diamond i(g)$ is in the same equivalence class as $i(f \circ g)$. In the construction of the equivalence relation in step 3 we require $f \diamond g \sim f \circ g$.

□

As discussed in section 2.2, an algebraic structure (X, ev_X) of type $(\mathcal{O}, \Lambda)$ is a functor sending 1 to X and the morphisms added in step 2 to their counterparts in $ev_X(\mathcal{O})$.

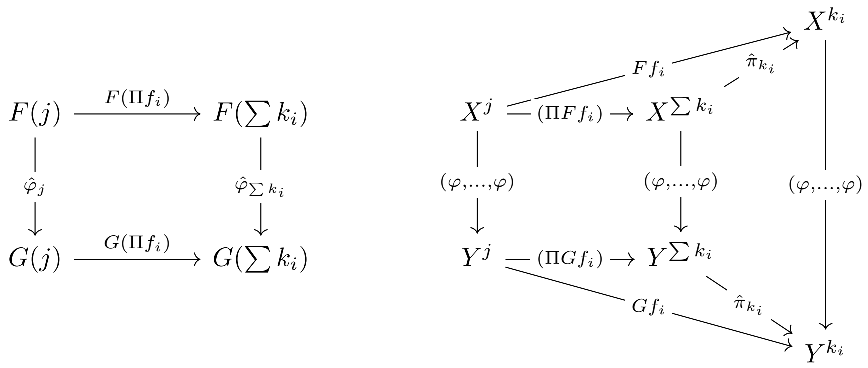
Theorem 2.6. *Recall that* $\mathsf{A}_{\mathcal{O},\Lambda}$ *is the category of algebraic structures of type* $(\mathcal{O}, \Lambda)$ *in which morphisms* $(X, ev_X) \to (Y, ev_Y)$ *are homomorphisms. Let* T *be the corresponding template category according to the construction above.*
Let $\Phi : \mathsf{A}_\mathsf{T} \to \mathsf{A}_{\mathcal{O},\Lambda}$ *be the functor that sends a functor* $F \in \mathsf{A}_\mathsf{T}$ *to the pair* $(F1, ev_{F1})$*, where* $ev_{F1}(\Xi) = ev_{F1}(\mathcal{O})$*, and a natural transformation* $\varphi : F \implies G$ *to the map* φ_1*. Then,* Φ *establishes an equivalence of categories.*

An equivalence of categories is established by a full, faithful and essentially surjective functor. A functor is essentially surjective when every object in the codomain is isomorphic to an object in the image of the functors. Full and faithful represent surjectivity and injectivity on all homsets respectively.

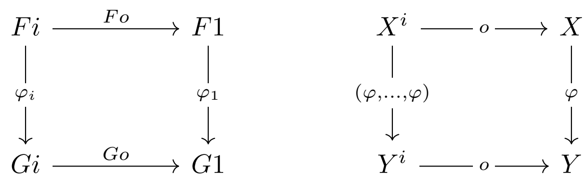
Proof. Recall from remark 3 that a product preserving functor F in A_T is uniquely determined from $F(1)$ and $F(\alpha)$ where $\alpha \in \mathcal{O}$ was added in the pretemplate.
As discussed in section 2.3, a natural transformation $\varphi : F \implies G$ corresponds to a homomorphism $F1 \to G1$ entirely determined by its component φ_1. Recall from section 1.2 that given another natural transformation $\psi : G \implies H$, the composition is given by $(\psi \circ \varphi)_X = \psi_X \circ \varphi_X$. More specifically, $\Phi(\psi \circ \varphi) = (\psi \circ \varphi)_1 = \psi_1 \circ \varphi_1 = \Phi(\psi) \circ \Phi(\varphi)$. Hence, Φ sends compositions of natural transformations to compositions of homomorphisms and is hence a functor.
As for every algebraic structure (X, ev_X) of type $(\mathcal{O}, \Lambda)$ we can find a product preserving functor sending 1 to X and the relevant operations to $ev_X(\mathcal{O})$, that satisfy the equations in Λ, Φ is essentially surjective.
Now fix two functors $F, G \in \mathsf{A}_\mathsf{T}$ and let $(X, ev_X), (Y, ev_Y)$ be their corresponding algebras.
To show Φ is full, let $\varphi : X \to Y$ be a homomorphism. We claim that $\hat{\varphi} : F \implies G$ whose components are $\hat{\varphi}_n = (\varphi, \ldots, \varphi) = \varphi^n$, the n-fold product of φ, is indeed a natural transformation. We only check the naturality condition for product morphisms and morphisms in Ξ, as single component projections $\hat{\pi}_\alpha$ where $\alpha : 1 \to i$ are

[8] Expressions in Λ here should be read as compositions of product morphisms. Recall that an expression is either an operation in $\mathcal{O}$, or a formal symbol of the form $e \circ (e_1, \ldots, e_n)$.

apparant, and naturality for compositions follows by glueing together two commutative squares.



The naturality condition for product morphisms (left) becomes clearer when we rewrite the square in terms of sets (right), with the appropriate projections added. By naturality of the f_i's, we have $\varphi^{k_i} \circ Ff_i = Gf_i \circ \varphi^j$, which by commutativity of the upper and lower triangles can be rewritten to $\varphi^{k_i} \circ \hat{\pi}_{k_i} \circ (\Pi Ff_i) = \hat{\pi}_{k_i} \circ (\Pi Gf_i) \circ \varphi^j$. Focussing now on the set of k_i'th coordinates, we see the inner square commmutes when fixing the k_ith components of the morphism between $X^{\sum k_i}$ and $Y^{\sum k_i}$ to φ.



The naturality for the morphisms in Ξ requires that $o \circ \varphi^i = \varphi \circ o$, in other words for any $x_1, \ldots, x_i \in X$ one has $o(\varphi(x_1), \ldots, \varphi(x_i)) = \varphi(o(x_1, \ldots, x_i))$, which is satisfied by virtue of φ being a homomorphism.
Lastly, Φ is faithful as every natural transformation is fully determined by φ_1. If $\Phi\varphi = \Phi\psi$ for two natural transformations, then $\varphi_1 = \psi_1$ and therefore $\varphi = \psi$. □

Remark 6. *Even though the objects in $\mathcal{T}_1$ are pairwise non-isomorphic, the same thing cannot be said for the quotient category. For example, if Λ would impose the condition $\Delta \circ \pi_1 = id_2$, then the objects 2 and 1 would be isomorphic (Note that by definition $\pi_1 \circ \Delta = id_1$). In other words, an algebra X from this template would satisfy $(x, y) = (x, x)$ for all $x, y \in X$. This makes sense, as the only set X for which $X^2 \cong X$, is a singleton set. The conclusion $x = y$ for all $x, y \in X$ is therefore not contradictory.*

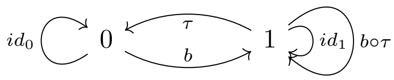
The construction of a template category with one generating object can be generalized to any template category, but we will omit its construction for notational sake. The key difference are the projection morphisms, which are given by $\hat{\pi}_{\alpha_1,\ldots,\alpha_m}$ projections on the m distinct coordinates.

3 Examples of finite lawvere algebras

In this section, we will provide written-out examples of template categories for well-established algebraic structures. Following the construction in 2.5, we omit most projections, compositions and product morphisms and only explicitly give some identity morphisms, terminal morphisms, and morphisms added in step 2, the latter of which we will refer to as generating morphisms.

3.1 Pointed Sets

Perhaps the simplest nontrivial algebraic structure is the pointed set, a set with a designated basepoint. Maps between pointed sets are often required to preserve the basepoint. Pointed sets often arise integrated in other structures, like pointed topological spaces which are needed for computing homotopy classes. Groups are also pointed sets, with the identity element as basepoint. The identity element is preserved under group homomorphisms.
Assigning the basepoint can be done by means of a nullary function into the set, and as such we want to find a template category with objects $0, 1$ and a nontrivial morphism $b : 0 \to 1$. Here is the template category T:



With the imposed condition that $\tau \circ b \circ \tau = \tau$ to ensure the map $\tau : 1 \to 0$ is unique.
Since this category is finite, we can verify all products manually. $0 \times 0 = 0$, because all morphisms into 0 are unique. To show $0 \times 1 = 1$ with the projections τ, id_1 we assign the following product morphisms:

$$id_0 \times b = b$$
$$\tau \times id_1 = id_1$$
$$\tau \times b \circ \tau = b \circ \tau$$

Since there is a unique map from every object to 0, the image of a functor from this template to Set is completely determined by whereto it maps 1 and b.
Maps between pointed sets are given by natural transformations $\alpha : F \implies G$ between two functors $F, G : \mathsf{T} \to \mathsf{Set}$. We can verify that these maps preserve the basepoint by naturality:

$$\begin{array}{ccc} F0 & \xrightarrow[Fb]{} & F1 \\ \downarrow{\scriptstyle \alpha_0} & & \downarrow{\scriptstyle \alpha_1} \\ G0 & \xrightarrow[Gb]{} & G1 \end{array}$$

Note that α_0 is a trivial map between two singleton sets.
Naturally, the concept of a pointed set can be generalized to a set with n basepoint by adding the appropriate amount of nontrivial morphisms $b_i : 0 \to 1$ to the template.

3.2 Magmas

Magmas are sets endowed with a binary operation.
Here is the template category, with only the generating morphisms:

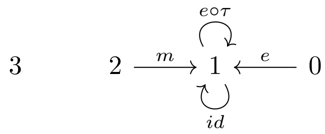
$$2 \xrightarrow{m} 1 \qquad 0$$

Note that this operation does not have to be associative by default. As an effect, the template category T is quite large and a prototypical morphism $f : 1 \to 1$ is a way of composing binary operation with itself and looks a bit like $f : x \mapsto m(m(x, m(x, x)), m(m(x, m(x, x)), m(x, x)))$.
Homomorphisms between magmas are again given by natural transformations, and are seen to preserve the magma law:

$$\begin{array}{ccc} F1 \times F1 & \xrightarrow{Fm} & F1 \\ \downarrow{\scriptstyle \alpha_2 = (\alpha_1, \alpha_1)} & & \downarrow{\scriptstyle \alpha_1} \\ G1 \times G1 & \xrightarrow{Gm} & G1 \end{array}$$

3.3 Monoids

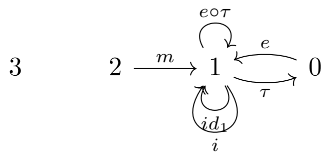
Magmas and pointed sets can be combined to make pointed magmas. If further restrictions are imposed on the magma law, we may obtain unital or associative magmas instead. A monoid combines those two, being defined as an associative unital magma.



Note that 3 appears as an object in this template, in order to give shape to associativity. Unitality is imposed by the condition $m \circ (id, e \circ \tau) = m \circ (e \circ \tau, id) = id$. Associativity is given by $m \circ (m, id) = m \circ (id, m)$.

3.4 Groups

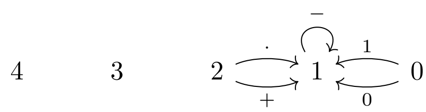
Groups are monoids with the extra inversion morphism.



It has the extra imposed conditions that $m \circ (id_1, i) = m \circ (i, id_1) = e \circ \tau$.

3.5 Rings

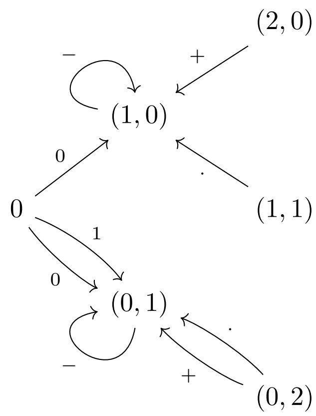
Rings have two basepoints, two binary operations and a unary operation associated with them:



besides the usual associativity, unitality and commutativity, invertability on the additive group, there is also some interplay between addition and multiplication through distributivity: $\cdot \circ (id, +) = + \circ (\cdot, \cdot) \circ (\pi_1, \pi_2, \pi_1, \pi_3)$. Note that the latter part uses a morphism $3 \to 4$ to duplicate the multiplicative scalar.

3.6 Modules

Modules have two underlying sets of interest, the module itself and the base ring, thus it has two generating objects.



The top part of the diagram corresponds to the underlying abelian group of the module, the bottom part corresponds to the base ring. Not all objects are shown for clarity, but the ring part is taken to the fourth power to model distributivity and the module part to the third power for associativity. Distributivity of scalar multiplication over module elements relies on the existence of the object $(2, 2)$.

Notably, in this definition of a module, the base ring is not fixed when considering module homomorphisms. If for two representations F, G the condition $F(0, 1) = G(0, 1)$ is imposed, and for a natural transformation α it is given that $\alpha_{(0,1)} = id_{F(0,1)} = id_{G(0,1)}$, the usual notion of a module homomorphism arises.

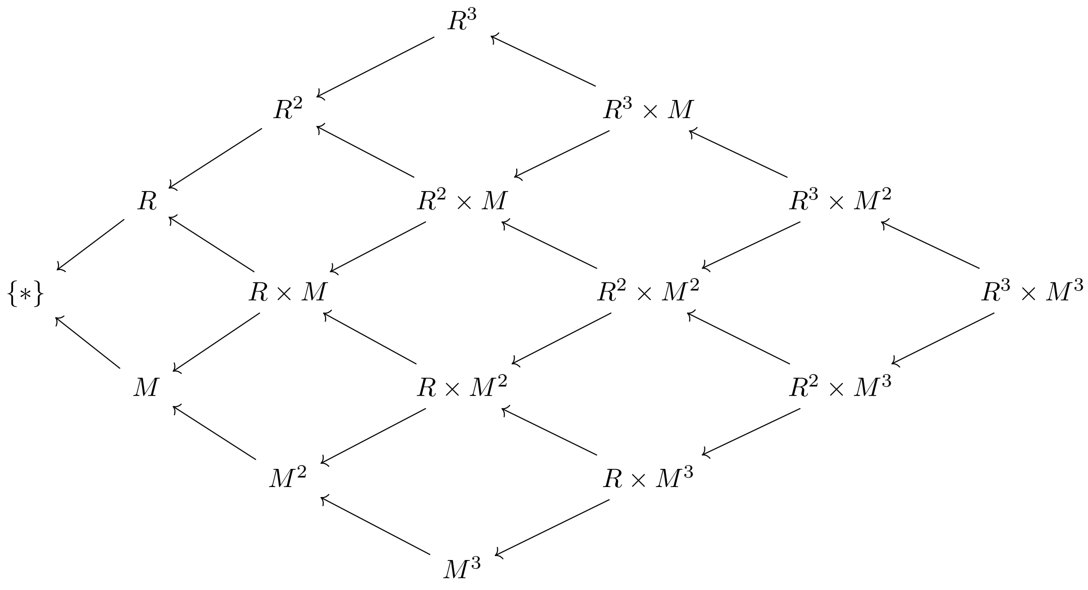


Figure 2: A coherence diagram for all the products needed to model modules. Not all projections are included. Note the existence of a singleton set acting as a terminal object. The object in the template category would be pairs (i, j) representing $R^i \times M^j$

3.7 Fields

With rings and modules succesfully modelled, one may attempt to specialize the results to fields and vector spaces. Unfortunately, a problem arises when modelling multiplicative inverses. An approach would be, just like for groups, to model multiplicative inverses with a global operation $i : 1 \to 1$. However, a representation from such a template could never yield a field, because the additive neutral element in a field is not invertible. This is further exemplified by the fact that so far all structures considered have products; given two representation functors $F, G : \mathsf{T} \to \mathsf{Set}$ it is possible to define a product $F \otimes G : \mathsf{T} \to \mathsf{Set}$ given by $(F \otimes G)(1) = F1 \times G1$ and $(F \otimes G)(f) = (Ff, Gf)$ for all morphisms. As an example from the template category of groups, $(F \otimes G)(m)$ would be the product function $(Fm, Gm) : (x_1, y_1), (x_2, y_2) \mapsto (Fm(x_1, x_2), Gm(y_1, y_2))$ which is precisely the induced group law on product groups. For fields, one of the most apparant reasons why no products exists is by the introduction of zero divisors, elements on which a global inversion morphism cannot be defined.

References

[BE72] Andrée Bastiani and Charles Ehresmann. Categories of sketched structures. *Cahiers Topologie Géom. Différentielle*, 13:105–214, 1972.

[Bra23] Martin Brandenburg. Large limit sketches and topological space objects, 2023.

[Lan78] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer Science+Business Media New York, 1978.

[Law63a] F. William Lawvere. Functorial semantics of algebraic theories. *Proc. Nat. Acad. Sci. U.S.A.*, 50:869–872, 1963.

[Law63b] F.W. Lawvere. Functorial semantics of algebraic theories (first notice). *Proc. Nat. Acad. Sci.*, (50):869–872, 1963.

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

[nla24] Lawvere theory in nlab. *available at* `https://ncatlab.org/nlab/show/Lawvere+theory`, 2024.

[SH00] Lutz Schröder and Horst Herrlich. Free adjunction of morphisms. *Appl. Categ. Structures*, 8(4):595–606, 2000.

[Tri18] Todd Trimble. multisorted lawvere theories. *available at url=https://ncatlab.org/toddtrimble/published/multisorted+Lawvere+theories, visited in February 2025*, 2018.