Contents

Chapter 1

Introduction

1.1 Problem

The Categorical Approach to Quantum Information [Abramsky and Coecke, 2004] heavily relies on so-called †-Frobenius algebras to represent Quantum Observables. One often wishes to work with several interacting observables. This can be achieved by specifying rules of interaction between the corresponding Frobenius algebras [**?**]. Furthermore, Frobenius algebras themselves consists of a monoid and a comonoid that interact via the Frobenius condition. There is an abundance of examples in mathematics, physics and computer science where simple theories are composed to give rise to more complicated ones.

To understand compositionality formally, one needs a formal system where a theory - or more precisely its specification - is a formal mathematical object and where one can specify interaction between two theories using tools of the this system. Universal algebra studies algebraic specifications as formal constructs. Categorical Universal Algebra (CUA) is its modern manifestation. In Categorical Universal Algebra, the specifications of a theory are usually expressed as either a category, a multi-category or a monad. One also has methods for expressing the *models* of a theory - the concrete examples of the formal theory - and for combining theories. For example, to specify the theory of groups, we create some structure G that captures the specifications of the theory. G contains information about the operations of the theory and its axioms. Concrete groups, such as $\mathbb{Z}_7$, for example, are objects in Set that are derived from G.

The most studied tool of CUA is the monad. A monad is a functor with additional data that captures specifications of a theory by encoding the recipe for creating free models. The monadic approach to CUA makes it easy to treat compositionality formally: Beck's distributive laws are an elegant method for combining monads. But the monadic approach also has several deficiencies. The first issue is that to specify a monad for a particular theory, one has to understand how the free models of that theory are constructed. This information is not always available. Second, a monad is a functor on a fixed base category. This means that all the models of the monad will be objects in that category. The implications of this are best seen via an example. Suppose that we wish to model the specifications for the theory of groups and then to study the models in both Set and Top. To study groups in the category of sets we need to specify a monad G : Set $\to$ Set and to study groups in the category of topological spaces and continuous maps between them - that is, to study topological groups - we need to specify a different monad G' : Top $\to$ Top. But both monads actually capture exactly the same operations and the same axioms.

The Operadic/PRO approach to CUA uses categories with structure - monoidal, symmetric monoidal or finite-product - to encode the specifications. The operations of the theory are encoded by morphisms in these categories and the axioms become commutative diagrams. One can also approach Categorical Universal Algebra via the theory of Operads. There multi-categories encode specifications for theories that have operations with multiple inputs, but only a single output. This approach decouples specifications from the models: models are structure preserving (multi) functors from the theory to a category of choice. So, once we have constructed a category G that encodes the theory of groups, the category of groups is equivalent to the category of all structure-preserving functors from G to Set, while the category of topological groups is equivalent to the category of all structure-preserving functors from G to Top. Since the morphisms of G are exactly the operations of the theory of groups, G contains morphisms that encode the inverse, multiplication and the unit. One can see these operations directly from the description of G. This makes it easier to build a category or a an operad for a particular specification.

The shift from monads to categories with structure has has two issues. First, unlike monads, there are no natural methods for combining categories. Thus this flexible framework does not naturally aid in the development of a general theory of compositionality. Second, there are many variants of categories with structure, each existing independently of the other ones. Operations in algebraic specifications have an arity. These arities in two specifications can have very different types. For example, some operations accept lists of elements, other accept finite sets while others accept infinite sets. Generally there is a different variant of a given CUA tool for each arity type. For example, plain operads capture theories that have linear arities. To model specifications for theories with linear symmetric arities, one has to use symmetric operads. Lawvere theories are designed to model theories with Cartesian arities.

Fortunately it was recently observed that both these problems can be solved by expressing tools of the Operadic/PRO approach as *internal monads* in the bicategory of profunctors. Two approaches have been proposed. The "Cambridge" approach, pioneered by Hyland (cf. [Hyland, 2010]), develops a framework in which the notion of type is a precise mathematical object. Then one can construct a profunctor-based framework where theories of a particular type are internal monads. In the second approach (cf. [Lack, 2004]), Lack shows that PRO-based structures are internal monads in a variant of Prof. Since both PROs and Operads are expressed as monads inside bicategories, they can be combined using internal distributive laws in these bicategories.

In Lack's work, one can express all theories with linear arities. That is, PROs, PROPs, PROBs and Lawvere theories. Regrettably, his framework does not accommodate more elaborate arities. On the other hand, Hyland's framework gives a generous definition of type that captures all the linear examples but is not limited to them. In this work, we mainly focus on extending Hyland's framework, but we often draw parallels with Lack's approach.

Hyland's framework draws its strength from its ability to express operads of any type T. It can do this because the notion of type is itself a formal mathematical object. The type T is actually a 2-monad on Cat that distributes over the presheaf monad $P : \mathsf{Cat} \to \mathsf{Cat}$. Because of this distributivity, the monad T can be extended to the Kleisli bicategory of P, which happens to be the bicategory of profunctors. This extension gives rise to a new monad $\tilde{T} : \mathsf{Prof} \to \mathsf{Prof}$. Theories of type T are internal monads in $\mathsf{kl}(\tilde{T})$. The dependence of $\mathsf{kl}(\tilde{T})$ on the type T allows to express a theory of any type in this framework.

Hyland's approach, creates, for each T, a framework $\mathsf{kl}(\tilde{T})$ in which theories of type T can be expressed. The only issue we have with it is that these frameworks are independent of each other. Given a type S and a type T, we express theories of type S as monads in $\mathsf{kl}(\tilde{S})$ and theories of type T as monads in $\mathsf{kl}(\tilde{T})$. In practice, S and T are often related. For example, the monad that corresponds to linear arities is a sub-monad of the monad that corresponds to symmetric linear arities. This indicates that there should be a relationship between theories - expressed as internal monads - with linear arities and symmetric linear arities.

In this thesis we extend Hyland's approach to take relationships between types into account. In particular we show that (a) a relationship between types is itself a formal mathematical object, (b) any relationship between types gives rise to a relationship between theories of these types. In other words, we construct a passage $\mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T})$ from a relationship $S \to T$. This passage sends *theories to theories* and *distributive laws to distributive laws*.

1.2 Outline

We now briefly describe each chapter in this work. Chapters 7, 8 and 9 are the main research chapters.

Chapter 2 gives a single-dimensional overview of the main results of the thesis. Its purpose is to give the reader an easy-to-follow intuitive view of the whole work, without all the technical details that are needed in the full 2-dimensional treatment. We introduce *binary theories*, which are toy theories where for each arity a there either exists an operation of arity a, or there are no such operations. Binary theories cannot actually specify what the operations are. While binary theories in themselves are not very useful, they do serve as a good introduction to Operads and to the idea of capturing types by using monads. We also show how relationships between types give rise to relationships between binary theories of these types.

Chapter 3 is an introduction to Categorical Universal Algebra. The chapter begins with a general discussion of theories. It then introduces three flavours of operads - plain, symmetric and Cartesian - and shows how these tools can express theories of the corresponding types. In particular, we give examples of concrete operads and their models. We discuss equivariance for symmetric operads in some detail. This is a fairly technical issue that complicates the definition of operads. In the abstract framework of Chapter 6, equivariance is hidden deep inside the definition, making operads more accessible. We also discuss PROs, PROPs and Lawvere Theories. There again we provide many examples to illustrate these concepts. We pay particular attention to Lawvere theories, where we go beyond the definition and state several important facts about these structures. The Cartesian structure of Lawvere theories makes them especially attractive for CUA. We finish with a comparison of these structures.

Chapter 4 introduces 2-Category Theory. We will need 2-Category Theory to construct a framework in which the tools of Chapter 3 can be efficiently expressed, combined and studied. We start with the familiar definitions of a 2-category and a bicategory and then provide impotent examples of each. Here we first define the bicategory Prof which is the basis for most of the constructions in this thesis. Once we understood 2-categories, we need to understand how thy can be compared and related. This is done via 2-functors. After the main technical definition we list important examples that we will carry through the rest of the work. 2-natural transformations and modifications are also defined. As we have done before, we give important examples of both constructs. The chapter concludes with a brief discussion of adjointness for 2-categories and monoidal 2-categories.

Chapter 5 deals with the theory of 2-monads and 2-distributive laws. In Chapter 4 we gave examples of several 2-functors. In this chapter we show that these 2-functors actually are part of 2-monads on Cat. Specifically, we study the monoidal category monad, the symmetric monoidal category monad, the Cartesian category monad and the presheaf monad on Cat. We discuss Kleisli bicategories and show that the Klesili bicategory of P is equivalent to Prof. The second part of the chapter focuses on 2-distributive laws. We begin with the simpler case when we distribute a functor over a monad. We note that Beck's Theorem for distributive laws lifts from single to 2-dimensional monad theory. To make the abstract theory more accessible, we present the 2-distributive law of the monoidal category functor over the presheaf monad. Monads can also be distribute over monads. We briefly skim over the general theory, and jump straight into an analysis of our main examples. We show that the monoidal category monad, the symmetric monoidal category monad and the Cartesian category monad all distribute over the presheaf monad. We give explicit formulas for the distributive laws and then compute the formulas for the actions of the extensions on profunctors.

Chapter 6 combines ideas developed in Chapters 3, 4 and 5. It shows how the 2-categorical tools of Chapters 5 and 6 give rise to a framework which allows for an elegant expression of the CUA tools of Chapter 3. The tools of Chapter 3 are either categories with structure or multi-categories with structure. Keeping this in mind, we begin by recalling that categories are internal monads in the bicategory of Spans. We then extend this result to show that profunctors can be used to add new categorical structure to an existing category. This observation is essential for understanding why PROs and PROPs can be viewed as monads in Prof(Mon). The compositional structure of multicategories is complicated. The fact that each morphism has several inputs leads to a tree-like composition structure that cannot be easily described in Prof. This is where the theory of Kleisli bicategories of Chapter 5 comes in. Internal monads inside Kleisli bicategories are exactly multicategories. This fact allows us to express operads, symmetric operads and Cartesian operads as monads in certain bicategories. We provide detailed proofs. While the results were known, the author has not seen them or their proofs anywhere else in literature. Finally, we give examples of concrete Operadic- and PRO/PROP theories. We also explore the idea of using internal distributive laws for combining theories and provide some important examples.

Chapter 7 is the first research chapter of the thesis. It formally develops a theory of extensions. To do this, we define two monoidal bicategories. The first one, DL^P, consists of distributive laws over P, 2-natural transformations that also distribute over P and modifications that interact with P in a coherent manner. Monads in DL^P are types of theories and monad morphisms in DL^P are relationships between such types. The second monoidal bicategory is the bicategory of extensions: Ext_P. Its 0-cells are functors on a bicategory $\mathcal{K}$ that can be extended to $\mathsf{kl}(P)$; 1-cells are 2-natural transformations that can be extended and 2-cells are extendible modifications. We define two 2-functors $\tilde{(-)} : \mathsf{DL}^P \to \mathsf{Ext}_P$ and $d : \mathsf{Ext}_P \to \mathsf{DL}^P$. The first major contribution of this thesis is *the pair* $\tilde{(-)}$ *and* d *forms a monoidal equivalence of 2-categories*. We obtain two important corollaries from this. The first one states that distributive laws of *monads over* P are mapped to internal monoids in the monoidal bicategory $\mathsf{Ext}(P)$. The second corollary shows that monoid morphisms in DL^P map to monoid morphism in Ext_P. Combined these corollaries give us a way of extending not only types $S : \mathsf{Cat} \to \mathsf{Cat}$, but also relationships between types.

Chapter 8 is the second research chapter of the thesis. In this chapter we prove two theorems. The first theorem shows that any monad morphism $\alpha : S \to T$ between two 2-monads gives rise to a 2-functor $\mathsf{kl}(S) \to \mathsf{kl}(T)$ between the associated Kleisli bicategories. 2-functors preserve internal monads and internal distributive laws. Thus, combined with the results of Chapter 7, this theorem shows how to build relationships between theories from a relationship between types. The second theorem explores a relationship in the other direction. When P is the presheaf monad, one can explore the duality $\mathsf{kl}(P) = \mathsf{Prof}$ to see that every component i_X of a 2-natural transformation $i : S \to T$ can be mapped to two profunctors: $(i_X)_*$ and $(i_X)^*$. Chapter 7 deals with $(i_X)_*$. In some cases $(i_X)^*$ is also a 2-natural transformation, even a monad morphisms (e.g. for some inclusions of 2-monads). In that case we obtain a 2-functor $\mathsf{kl}(T) \to \mathsf{kl}(S)$. The second theorem shows that under mild coherence conditions the functors obtained from i form a lax 2-adjunction. When i is the inclusion of one type into another, the results of this Chapter have a familiar interpretation: the functor obtained from i_* gives rise to a 2-functor that creates theories of type T from theories of type S freely. The functor obtained from i^* is the forgetful functor. It sends theories of type T to theories of type S by forgetting some of the arity structure.

Chapter 9 applies the theory developed in Chapters 7 and 8 to important CUA examples. First we examine the monad morphism $i : M \to S$ that is at each component just the inclusion of a free monoidal category into a free symmetric monoidal category. We show in great detail that the inclusion satisfies the various coherence conditions needed for the application of Theorem 1 of Chapter 8. This gives a 2-functor from $\mathsf{kl}(M)$ to $\mathsf{kl}(S)$ that shows how to build symmetric operads from plain ones freely. This 2-functor also sends distributive laws between operads to distributive laws between the associated free symmetric operads. We generate an explicit formula for this functor. Theorem 2 also applies to i. Thus a left-adjoint $J \vdash I : \mathsf{kl}(M) \to \mathsf{kl}(S)$ also exists. We perform the same analysis on the inclusion $j : S \hookrightarrow F$ of symmetric monoidal categories into Finite-product categories. The inclusion also gives a lax-adjoint pair. The latter pair shows how to construct Lawvere theories freely from symmetric operads. Crucially, it also shows that distributive laws between the symmetric operads are mapped to distributive laws between the associated free Lawvere theories.

Chapter 10 is the conclusion chapter. In this Chapter we give a summary of our main results and briefly explore several new research questions in the field.

Chapter 2

Single dimensional overview

The main results of this thesis are formulated in the language of 2-Dimensional Category Theory. But before delving into all the technical machinery that accompanies higher dimensions, we explore single dimensional analogues of these results. The purpose of this is to provide the reader with a simple high-level overview of the main tools we employ and our main contributions. We note that in this simplified environment many results appear trivial, but consequently they aid in developing an intuition for the work as a whole.

We begin by defining single dimensional analogues of operads (see Chapter 2). We call these tools *binary theories*. We then develop a framework in which we can discuss binary theories of various types, emphasising that the notions of 'type' is a mathematical object in itself.

We conclude this chapter with an analysis of how relationships between types lead to relationships between theories of these types. This last part is the main contribution of the thesis and is covered in full detail in Chapters 7, 8 and 9.

2.1 Binary Theories

A *binary theory* models the scenario where for each arity, there is at most one operation. We model this by defining a theory to be a function t with codomain $2 = \{0, 1\}$. We interpret $t(x) = 1$ to mean that there exists an operation of arity x. $t(x) = 0$ means that there is not such operation.

There is a definition of a binary theory for many different types of arities. We begin with the simplest case: when the arities are linear.

A *linear binary theory* consists of a function $t : \mathbb{N} \to \{0, 1\}$, such that $t(1) = 1$ and for each $k \in \mathbb{N}, n_1, \ldots, n_k \in \mathbb{N}$

$$\big(t(k) = 1 \text{ and } \forall\, 1 \leq i \leq k \;\; t(n_i) = 1\big) \implies t(n_1 + \ldots + n_k) = 1. \tag{2.1}$$

The requirement $t(1) = 1$ says that the theory always contains the identity element. The second condition makes the theory compositional: k operations of arities n_i can be "plugged into" an operation of arity k to produce a new operation of arity $\sum k_i$.

Concretely, linear binary theories are subsets of $\mathbb{N}$ with certain properties. The subset $\{1\} \subseteq \mathbb{N}$ is the simplest binary theory. The subset $\{1 + 2k \mid k \geq 0\} \subseteq \mathbb{N}$ is a linear binary theory generated by a single operation of arity 3.

We can also study binary theories whose types are lists with polarity. This feature is modelled by adding an indication '$-$' or '$+$' to each arity to indicate the polarity of the list. Formally, these theories are functions $t : \mathbb{Z} \to 2$ that satisfies Equation 2.1 and $t(1) = 1$.

2.2 Monads

The central mathematical object in this thesis will be the monad. A *monad* on a category $\mathcal{C}$ consists of an endofunctor $T : \mathcal{C} \to \mathcal{C}$ and natural transformations $m : T^2 \to T$ and $u : 1_{\mathcal{C}} \to T$ making

$$\begin{tikzcd} T^3 \arrow[r, "mT"] \arrow[d, "Tm"'] & T^2 \arrow[d, "m"] \\ T^2 \arrow[r, "m"'] & T \end{tikzcd} \qquad \begin{tikzcd} T \arrow[r, "uT"] \arrow[dr, equal] & T^2 \arrow[d, "m"] & T \arrow[l, "Tu"'] \arrow[dl, equal] \\ & T & \end{tikzcd}$$

commute.

We recall important examples of monads.

Example 2.1. The monoid monad $M : \mathsf{Set} \to \mathsf{Set}$ sends every object to the monoid $MX = (MX, *, e)$. The set MX consists of all the words on the alphabet X,

$$[x_1, \ldots x_n] * [y_1, \ldots y_m] = [x_1, \ldots, x_n, y_1, \ldots y_m].$$

and $e = []$, is the empty word.

The multiplication on $m : M^2X \to MX$ sends a list of words

$$\big[[x_{11}, \ldots x_{1n_1}], \ldots, [x_{k1}, \ldots, x_{kn_k}]\big]$$

to the word formed by removing the inner brackets: $[x_{11}, \ldots x_{kn_k}]$. The unit $u : X \to MX$ sends each element x to the single letter word $[x] \in MX$.

Example 2.2. The *powerset monad* on Set sends every set X to its powerset PX. Multiplication is given by

$$m_X : PPX \to PX, \quad \mathcal{A} \to \bigcup \mathcal{A}$$

and the unit sends every $x \in X$ to the singleton $\{x\} \in PX$.

Monads provide concise semantics for various mathematical and informatic constructs. For example the monoid monad encodes all the information needed to recover the category of monoids and monoid homomorphisms. Each monad gives rise to two categories of algebras: its category of *Eilenberg-Moore Algebras* and its *Kliesli category*.

The *Eilenberg-Moore category* of a monad T on $\mathcal{C}$, written $\mathcal{C}^T$, has as objects $\mathcal{C}$-morphisms $h : TX \to X$ such that

$$
\begin{tikzcd}
T^2X \arrow[r, "Th"] \arrow[d, "m_X"'] & TX \arrow[d, "h"] \\
TX \arrow[r, "h"'] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "u_X"] \arrow[dr, equal] & TX \arrow[d, "h"] \\
 & X
\end{tikzcd}
\tag{2.2}
$$

These objects are called the *algebras* of T. A *homomorphism of algebras* is a $\mathcal{C}$-morphism $f : X \to Y$ such that

$$
\begin{tikzcd}
TX \arrow[r, "Tf"] \arrow[d, "h"'] & TY \arrow[d, "k"] \\
X \arrow[r, "f"] & Y
\end{tikzcd}
$$

commutes. Composition and units are imported from $\mathcal{C}$.

The *Kliesli category*, denoted $\mathcal{C}_T$, has the same objects as $\mathcal{C}$; the morphisms are given by $\mathcal{C}_T(X,Y) = \mathcal{C}(X,TY)$. The identity on X is the component u_X of the unit natural transformation. Given morphisms $\varphi : X \to Y$ and $\psi : Y \to Z$, we compose them in $\mathcal{C}_T$ by taking the following composite in $\mathcal{C}$

$$X \xrightarrow{\varphi} TY \xrightarrow{T\psi} T^2Z \xrightarrow{m_Z} TZ.$$

Example 2.3. The category Set^M of Eilenberg-Moore algebras for the monoid monad is isomorphic the category Mon on monoids and monoid homomorphisms. (cf. Example 20.5(2) of [Adamek et al., 1990]).

Example 2.4. P-algebras are complete lattices. The algebra mapping $h : PX \to X$ equips each subset of X with a choice of supremum in X.

Example 2.5. The Kleisli category of P is the category of sets and relations. Indeed, $\mathsf{kl}(P)(X,Y) = \mathsf{Set}(X,PY) \simeq \mathsf{Rel}(X,Y)$. There are two important identity-on-objects functors from Set to Rel.

$$(-)_* : \mathsf{Set} \to \mathsf{Rel}, \quad (f : X \to Y) \mapsto (f_* : X \to Y), \quad x(f_*)y \iff f(x) = y$$

and

$$(-)^* : \mathsf{Set}^{\mathrm{op}} \to \mathsf{Rel}, \quad (f : X \to Y) \mapsto (f^* : Y \to X), \quad y(f^*)x \iff f(x) = y$$

We will often compare monads. Let (T,m,u) be a monad on $\mathcal{C}$ and (S,n,v) a monad on $\mathcal{D}$. A *morphism of monads* from (T,m,u) to (S,n,v) is a pair (A,a) with $A : \mathcal{C} \to \mathcal{D}$ a functor and $a : SA \to AT$ a natural transformation satisfying

$$
\begin{tikzcd}
A \arrow[r, "uA"] \arrow[dr, "Av"'] & SA \arrow[d, "a"] \\
 & AT
\end{tikzcd}
\qquad
\begin{tikzcd}
SSA \arrow[r, "Sa"] \arrow[d, "nA"'] & SAT \arrow[r, "aT"] & ATT \arrow[d, "Am"] \\
SA \arrow[rr, "a"'] & & AT.
\end{tikzcd}
$$

Proposition 2.6. *Let $(A,a) : (T,m,u) \to (S,n,v)$ be a morphism of monads. Then there is a functor $\hat{A} : \mathcal{C}^T \to \mathcal{D}^S$ sending every T-algebra $h : TX \to X$ to the S-algebra $\hat{A}h :=\ SAX \xrightarrow{a_X} ATX \xrightarrow{Ah} AX$*

and an algebra homomorphism $f : (X,h) \to (Y,k)$ to Af. Furthermore, the following diagram commutes:

$$\begin{tikzcd} \mathcal{C}^T \arrow[r, "\hat{A}"] \arrow[d, "U"'] & \mathcal{D}^S \arrow[d, "U"] \\ \mathcal{C} \arrow[r, "A"'] & \mathcal{D}. \end{tikzcd}$$

Proof. $\hat{A}h$ is indeed an algebra:

$$\begin{aligned} \hat{A}h \cdot m_{AX} = Ah \cdot a_X \cdot m_{AX} &= Ah \cdot An_X \cdot a_{TX} \cdot Sa_X \\ &= Ah \cdot ATh \cdot a_{TX} \cdot Sa_X \\ &= Ah \cdot a_X \cdot SAh \cdot Sa_X = \hat{A}h \cdot S\hat{A}h. \end{aligned}$$

Similarly one checks that $\hat{A}f$ is an algebra homomorphism.

The commutativity of the diagram is immediate from the definition of $\hat{A}$ as is its functoriality. □

The functor $\hat{A}$ in the proposition is called the *lift* of A along the forgetful functor U.

The reader may have noticed that we have made an explicit choice in the direction of the natural transformation a in the definition of a monad morphism. Nothing is stopping us from choosing the natural transformation to go in the other direction: $b : AT \to SA$. And we make this choice in the following definition. Let (T,m,u) be a monad on $\mathcal{C}$ and (S,n,v) a monad on $\mathcal{D}$. A *opmorphism of monads* from (T,m,u) to (S,n,v) is is a pair (A,b) with $A : \mathcal{C} \to \mathcal{D}$ a functor and $b : AT \to SA$ a natural transformation satisfying

$$\begin{tikzcd} A \arrow[r, "Av"] \arrow[dr, "uA"'] & AT \arrow[d, "b"] \\ & SA \end{tikzcd} \qquad \begin{tikzcd} ATT \arrow[r, "bT"] \arrow[d, "Am"'] & SAT \arrow[r, "Sb"] & SSA \arrow[d, "nA"] \\ AT \arrow[rr, "b"'] & & SA. \end{tikzcd}$$

Proposition 2.7. *Let $(A,b) : (T,m,u) \to (S,n,v)$ be a opmorphism of monads. Then there is a functor $\tilde{A} : \mathcal{C}_T \to \mathcal{D}_S$ that coincides with A on objects. On morphisms*

$$\tilde{A}(\ X \xrightarrow{\varphi} TY\ ) = \ AX \xrightarrow{A\varphi} ATY \xrightarrow{b_Y} SAY\ .$$

Furthermore, the following diagram commutes:

$$\begin{tikzcd} \mathcal{C} \arrow[r, "A"] \arrow[d, "(-)_*"'] & \mathcal{D} \arrow[d, "(-)_*"] \\ \mathcal{C}_T \arrow[r, "\tilde{A}"'] & \mathcal{D}_S. \end{tikzcd}$$

Proof. We verify that $\tilde{A}$ is functorial. For $\psi : X \nrightarrow Y$ and $\varphi : Y \nrightarrow Z$ we have

$$\begin{aligned}
\tilde{A}\psi \cdot \tilde{A}\varphi &= n_{AZ} \cdot Sb_Z \cdot SA\varphi \cdot b_Y \cdot A\psi \\
&= n_{AZ} \cdot Sb_Z \cdot b_{TZ} \cdot AT\varphi \cdot A\psi \\
&= b_Z \cdot Am_Z \cdot A(T\varphi \cdot \psi) = \tilde{A}(\varphi \cdot \psi).
\end{aligned}$$

Preservation of identities is immediate from the triangle in the definition of a opmorphism of monads.

It remains to verify that the square in the Proposition commutes. All four functors are identity-on-objects. On morphisms,

$$\begin{aligned}
(\tilde{A} \cdot (-)_*)f &= \tilde{A}(\ X \xrightarrow{f} Y \xrightarrow{v_Y} TY\ ) \\
&= AX \xrightarrow{Af} AY \xrightarrow{Av_Y} ATY \xrightarrow{b_Y} SAY \\
&= AX \xrightarrow{Af} AY \xrightarrow{u_{AY}} SAY\ = (Af)_*.
\end{aligned}$$

□

We call $\tilde{A}$ the *extension* of A.

Example 2.8. The pair $(1_\mathcal{C}, u)$ where u is the unit of the monad (T, m, u), is a morphism of monads $(1_\mathcal{C}, 1, 1) \to (T, m, u)$. The monad morphism axioms follow from the monad axioms for T.

Example 2.9. Let M and G be the monoid and group monads on Set, respectively. Then $(1_{\mathsf{Set}}, i)$, with $i_X : MX \to GX$ the inclusion, is a morphism of monads from G to M. It induces a functor

$$I : \mathsf{Grp} = \mathsf{Set}^G \to \mathsf{Set}^M = \mathsf{Mon}$$

that sends

$$(GX \to X) \mapsto (MX \to GX \to X),$$

so that I just forgets the inverses of X.

2.3 Distributive Laws

Given two monads $T, S : \mathcal{C} \to \mathcal{C}$ in general ST is no longer a monad. For example, there is no canonical way to define the multiplication

$$STST \to ST.$$

A natural transformation $TS \to ST$, would provide a candidate for multiplication

$$STST \to SSTT \to ST.$$

A distributive law is such a natural transformation, which satisfies the right conditions for the resulting morphisms to become a multiplication. More precisely a *distributive law between S and T* is a natural transformation

$$l : TS \to ST$$

making

$$\begin{tikzcd} & T \arrow[dl, "T\eta_S"'] \arrow[dr, "\eta_S T"] & \\ TS \arrow[rr, "l"] & & ST \\ & S \arrow[ul, "\eta_T S"] \arrow[ur, "S\eta_T"'] & \end{tikzcd} \qquad \begin{tikzcd} TSS \arrow[r, "lS"] \arrow[d, "T\mu_S"'] & STS \arrow[r, "Sl"] & SST \arrow[d, "\mu_S T"] \\ TS \arrow[rr, "l"] & & ST \\ TTS \arrow[u, "\mu_T S"] \arrow[r, "Tl"'] & TST \arrow[r, "lT"'] & STT \arrow[u, "S\mu_T"'] \end{tikzcd} \tag{2.3}$$

commute.

A distributive law gives rise to the *composite monad*

$$(ST, \eta_S T \cdot \eta_T, S\mu_T \cdot \mu_S TT \cdot SlT).$$

There is also a notion of a distributive law between a monad (T, m, u) and a functor S. In this case, we only require the bottom rectangle and triangle of 2.3 to commute. Another view point on such distributive laws is afforded by the notion of monad morphisms.

Proposition 2.10. *A distributive law between a monad (T, m, u) and a functor S is a natural transformation $\lambda : TS \to ST$ such that $(S, \lambda) : (T, m, u) \to (T, m, u)$ is a morphism of monads.*

Example 2.11. The canonical example of a distributive law is the distributivity of multiplication over addition in a ring: $x \cdot (y + z) = x \cdot y + x \cdot z$. This simple interaction between two algebraic structures can be reconstructed from a distributive law of the monad for Abelian groups, A, over the monoid monad M. The assignment

$$\begin{aligned} &(x_1 + \ldots + x_n) \cdot (y_1 + \ldots + y_{n'}) \cdot \ldots \cdot (z_1 + \ldots + z_{n''}) \\ &\mapsto x_1 \cdot y_1 \cdot \ldots \cdot z_1 + x_1 \cdot y_1 \cdot \ldots \cdot z_2 + \ldots + x_n \cdot y_n \cdot \ldots \cdot z_n \end{aligned}$$

defines the component λ_X of the distributive law $\lambda : MA \to AM$. This distributive law turns AM into a monad whose algebras are rings.

Example 2.12. Another example, and one that will reappear throughout this thesis, is the distributivity of the monoid monad over the powerset monad. The X component of the distributive law, $\lambda_X : MPX \to PMX$ is given by

$$[A_1, \ldots A_n] \mapsto \{[a_1, \ldots a_n] \mid a_i \in A_i\}. \tag{2.4}$$

The importance of this particular distributive law compels us to verify that it indeed satisfies the axioms.

Proposition 2.13. *The assignment in (2.4) defines a distributive law.*

Proof. We start by verifying that, with T replaced by M and S replaced by P, the upper rectangle of 2.3 commutes. The upper-right passage sends

$$
\begin{array}{rcl}
[\mathcal{A}_1, \ldots, \mathcal{A}_n] & \mapsto & \{[A_1, \ldots, A_n] \mid \forall i = 1 \ldots n \quad A_i \in \mathcal{A}_i\} \\
& \mapsto & \{P\lambda_X([A_1, \ldots, A_n]) \mid A_i \in \mathcal{A}_i\} \\
& = & \{\{[a_1, \ldots, a_n]\} \mid a_i \in A_i\} \mid A_i \in \mathcal{A}_i\} \\
& \mapsto & \bigcup_{A_i \in \mathcal{A}_i} \{\{[a_1, \ldots, a_n] \mid a_i \in A_i\} \\
& = & \{[a_1, \ldots, a_n] \mid \forall i \exists A_i \in \mathcal{A}_i \text{ such that } a_i \in A_i\}.
\end{array}
$$

and this coincides with the bottom-passage:

$$
\begin{array}{rcl}
[\mathcal{A}_1, \ldots, A_n] & \mapsto & [\bigcup \mathcal{A}_1, \ldots, \bigcup \mathcal{A}_n] \\
& \mapsto & \{[a_1, \ldots, a_n] \mid a_i \in \bigcup_i \mathcal{A}_i\}.
\end{array}
$$

The bottom rectangle of 2.3 also commutes. Both passages send $\Big[[A_{11}, \ldots, A_{1n_1}], \ldots, [A_{m1}, \ldots, A_{mn_m}]\Big]$ to $\{[a_{11}, \ldots, a_{1n_1}, \ldots, a_{m1}, \ldots, a_{mn_m}] \mid a_{ij} \in A_{ij}\}$.

The commutativity of the triangles is an easy exercise.

□

Distributive laws allow us not only to form the composite monad ST, but also to lift T to the Eilenberg-Moore category of S, and to extend T to the Kliesli category of S. In fact, all those concepts are equivalent.

Theorem 2.14 (Beck). *Let (T, m, u) and (S, n, v) be monads on a category $\mathcal{C}$. There is a one-to-one correspondence between the following concepts:*

1. *A distributive law $\lambda : TS \to ST$;*
2. *A multiplication $m : STST \to ST$ such that $(ST, m, u_S u_T)$ is a monad, the natural transformations*

$$S \xrightarrow{Su_T} ST \xleftarrow{u^S T} T$$

are monad morphisms and the middle unitary law

$$
\begin{tikzcd}
& ST \arrow[dl, "Su_T u_S T"'] \arrow[dr, equal] & \\
STST \arrow[rr, "m"'] & & ST
\end{tikzcd}
$$

holds;

3. *Liftings $\hat{T}$ of T to C^S;*
4. *Extensions $\tilde{S}$ of S to C_T.*

Proof. We sketch the proof of (1) $\iff$ (3) and refer the reader to [Beck, 1969] for the rest.

(1) $\implies$ (3): Define $\tilde{T}(\; SX \xrightarrow{h} X \;) = \; STX \xrightarrow{\lambda_X} TSX \xrightarrow{Th} TX$. The image of h under T an S-algebra since the diagram below commutes.

$$\begin{tikzcd}
S^2TX \arrow[r, "S\lambda_X"] \arrow[dd, "m_{TX}"'] & STSX \arrow[r, "STh"] \arrow[d, "\lambda_{SX}"'] \arrow[dr, phantom, "(2)"] & STX \arrow[d, "\lambda_X"] \\
{} \arrow[r, phantom, "(1)"] & TS^2 \arrow[r, "TSh"] \arrow[d, "Tm_X"'] \arrow[dr, phantom, "(3)"] & TSX \arrow[d, "Th"] \\
STX \arrow[r, "\lambda_X"'] & TSX \arrow[r, "Th"'] & STX
\end{tikzcd}$$

Indeed, (1) follows from one of the axioms for a distributive law, (2) follows from the naturality of λ and (3) commutes since h is an S-algebra.

On morphisms $\tilde{T} = T \cdot U$, with $U : \mathcal{C}^T \to \mathcal{C}$ the forgetful functor. The naturality of λ guarantees that this assignment sends algebra morphisms to algebra morphisms. The definition of $\tilde{T}$ immediately implies that is it in fact a lift.

(3) $\Longrightarrow$ (1): Define λ to be the composite

$$ST \xrightarrow{STv} STS = STUF = SU\tilde{T}F = UFU\tilde{T}F \xrightarrow{U\epsilon\tilde{T}F} U\tilde{T}F = TUF = TS \; .$$

The axioms are easy to check. $\square$

The distributive law in (2.4) allows us to extend M to the Kliesli category of P, i.e. to Rel. Given a relation $r : X \to PY$, Mr is given by

$$\tilde{M}(r)([x_1, \ldots, x_n]) = \{[y_1, \ldots y_n] \mid x_1ry_1, \ldots, x_nry_n\}.$$

We can also lift P to the Set^M:

$$\hat{P}(\; MX \xrightarrow{h} X \;) = \; PMX \xrightarrow{\lambda_X} MPX \xrightarrow{Ph} PX \; .$$

$\hat{P}(h)$ equips PX with the structure of a monoid; given $A, B \subseteq X$

$$\begin{aligned}
A * B &= \hat{P}(h)([A, B]) \\
&= Ph \cdot \lambda_X([A, B]) \\
&= Ph(\{[a, b] \mid a \in A, b \in B\} \\
&= \{h([a, b]) \mid a \in A, b \in B\} \\
&= \{a * b \mid a \in A, b \in B\}.
\end{aligned}$$

The lift $\hat{P}$ is now a monad on Mon. Its algebras are monoids that come equipped with complete lattice structure which satisfies

$$\bigvee (A * B) = \bigvee A * \bigvee B.$$

The above arguments work equality well when M is replaced by the monad for commutative monoids. In that case the algebras for the lift $\tilde{P}$ are the so-called *commutative quantales*. That is, complete lattices $(X, \bigvee)$

with a commutative monoid structure $*$ such that

$$x * \bigvee_{a \in A} a = \bigvee_{a \in A} x * a.$$

2.4 Monads in Kleisli 2-categories

The category Rel is Ord-enriched. This equips it with 2-categorical structure in which the 2-cells are given by set-inclusion. For any monad S : Rel $\to$ Rel, the Kleisli category kl(S) inherits the 2-categorical structure from Rel. We can thus study monads in such Kleisli categories (see Chapter 6 for more details on internal Category Theory).

We are particularly interested in exploring such monads inside the Kleisli 2-category kl$(\tilde{M})$ of the extension of the monoid monad to Rel. A monad on 1 in kl$(\tilde{M})$ consists of a relation $r : 1 \nrightarrow M1 = \mathbb{N}$ with

$$r * r \subseteq r \text{ and } (1_X)_* \subseteq r. \tag{2.5}$$

Equivalently, such a relation is a function

$$r : \mathbb{N} \to \{0, 1\}.$$

The second condition in Equation 2.5 implies that $r(1) = 1$.

The first condition is subtler. The composite $r * r$ in kl$(\tilde{M})$ equals to the following composite in Rel

$$1 \xrightarrow{r} \mathbb{N} \xrightarrow{\tilde{M}r} M\mathbb{N} \xrightarrow{\mu_{1*}} \mathbb{N}\ .$$

So $r * r(n) = 1$ if and only if

- there exists $k \in \mathbb{N}$ such that $r(k) = 1$ and
- there exists $n_1, \ldots n_s \in \mathbb{N}$ such that $[n_1, \ldots n_s](\tilde{M}r)k$ and
- $[k_1, \ldots, k_r](\mu_1)_* n.$

The last condition holds if, and only if, $k_1 + \ldots + k_r = n$. The second condition implies that $s = k$. Hence $r * r \subseteq r$ is the same as: for all $k \in \mathbb{N}$, $n_1, \ldots, n_k \in \mathbb{N}$

$$r(k) = 1 \text{ and } \forall\, 1 \le i \le k \;\; r(n_i) = 1 \implies r(n) = r(n_1 + \ldots + n_k) = 1.$$

We thus proved

Proposition 2.15. *Monads on 1 in* kl$(\tilde{M})$ *are exactly linear binary theories.*

The only thing that makes M special here is the fact that it distributes over P. In general, whenever T is a monad on Set that distributes over P, we define a *binary theory of type* T as an internal monad in the 2-category kl$(\tilde{T})$. So the *type of our theory* is encapsulated in the Set-monad T.

Example 2.16. Consider the Abelian group monad $A = (A, m, u)$ on Set. It too distributes over the powerset monad. The distributive law $\lambda : AP \to PA$ is given by

$$[-Y_1, \ldots, -Y_n, Y'_1, \ldots, Y'_m] \mapsto \{[-y_1, \ldots, -y_n, y'_1, \ldots, y'_m] \mid y_i \in Y_i, y'_j \in Y'_j\}.$$

Thus we can extend A to $\tilde{A} : \mathsf{Rel} \to \mathsf{Rel}$. Using the reasoning we used for M, we can compute the action of $\tilde{A}$ on a relation r:

$$[-x_1, \ldots, -x_k, x'_1, \ldots, x'_{k'}]\tilde{A}r[-y_1, \ldots - y_k, y'_1, \ldots y'_{k'}] \iff \forall i, j \quad x_i r y_i \text{ and } x'_j r y'_j.$$

Theories of type A are monads in the 2-category $\mathsf{kl}(\tilde{A})$. The arities of these theories are more expressive than arities of simple linear theories. The lists that act as inputs into our operations have two kinds of elements: negative and positive. Composition of operations now has to be mindful of the arities of operations.

2.5 Comparing Types of Theories

Fix two monads S and T and suppose that both distribute over P via distributive laws λ and ρ; in other words, each one corresponds to a type. Suppose that there is a monad morphism $\alpha : S \to T$. Can we then construct a relationship between $\mathsf{kl}(\tilde{S})$ and $\mathsf{kl}(\tilde{T})$ that assigns monads to monads?

Simply a functor $\mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T})$ will not suffice, since it does not have to send internal monads to internal monads. What we want is a 2-functor (see Chapter 4 for a detailed definition). 2-functors $A : \mathsf{Rel} \to \mathsf{Rel}$ are nothing but functors that preserve the order of hom-sets. That is

$$r \subseteq s \implies Ar \subseteq As$$

Its easy to see that such 2-functors preserve monads. Indeed,

$$r * r \subseteq r \implies Ar * Ar = A(r * r) \subseteq Ar.$$

We shall build the functor A in two steps. First, we will show that when the monad morphism α is compatible with the distributive laws λ and ρ, it too can be extended to

$$\tilde{\alpha} : \tilde{S} \to \tilde{T}.$$

Second, we show that post-composing morphism with this extension

$$(X \nrightarrow SY) \mapsto (X \nrightarrow SY \nrightarrow TY)$$

gives rise to the desired 2-functor. All this is made precise in the following two theorems:

Theorem 2.17 (Akhvlediani)**.** *Let T, S, P be monads on* Set *and let*

$$\lambda : TP \to PT \text{ and } \rho : SP \to PS$$

be distributive laws. Let $\alpha : S \to T$ be a natural transformation such that

$$\lambda \cdot \alpha P = P\alpha \cdot \rho \tag{2.6}$$

Then the formula

$$\tilde{\alpha}_X := (\alpha_X)_* = \; SX \xrightarrow{\alpha_X} TX \xrightarrow{u_{TX}} PTX \; = SX \nrightarrow TX.$$

gives rise to a natural transformation $\tilde{\alpha} : \tilde{S} \to \tilde{T}$. $\tilde{\alpha}$ is a monad morphism whenever α is.

Theorem 2.18 (Akhvlediani)**.** *Let $\tilde{S}$ and $\tilde{T}$ be monads on* Rel *and $\tilde{\alpha} : \tilde{S} \to \tilde{T}$ be a monads morphism. Then the assignment*

$$\varphi \mapsto \alpha_Y \cdot \varphi$$

defines a 2-functor

$$A : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T}).$$

We prove those theorems in chapters 7 and 8.

These two theorems can now be used to understand how the relationship between monoids and groups gives rise to a relationship between theories of type M and theories of type A. First, observe that the inclusion of a free monoid on X into the free group on X gives rise to a monad morphism

$$\iota : M \to A.$$

Naturality of ι is trivial. Since in both monads the multiplication is just concatenation of lists, ι is also a monad morphism. It is also easy to verify that Equation 2.6 holds for ι. Indeed, given $[Y_1, \ldots, Y_n] \in MPX$, the passages

$$[Y_1, \ldots, Y_n] \xmapsto{\iota_{PX}} [Y_1, \ldots, Y_n] \xmapsto{\rho_X} \{[y_1, \ldots y_n] \mid y_i \in Y_i\}$$

and

$$[Y_1, \ldots, Y_n] \xmapsto{\lambda_X} \{[y_1, \ldots y_n] \mid y_i \in Y_i\} \xmapsto{Pi_X} \{[y_1, \ldots y_n] \mid y_i \in Y_i\}$$

yield the same result.

Thus we can extend ι to $\tilde{\iota} : \tilde{M} \to \tilde{A}$. This extension is again a morphisms of monads. Thus, by Theorem 2.18 we obtain a 2-functor

$$I : \mathsf{kl}(\tilde{M}) \to \mathsf{kl}(\tilde{A}).$$

This 2-functor sends an M-theory $\varphi : X \nrightarrow MX$ to an A-theory $X \xrightarrow{\varphi} MX \xrightarrow{\iota_*} AX$. A quick look at the formula shows that the an operation in the new theory, $I\varphi$, exists if, and only if, it exists in the old theory φ.

Unfortunately things cannot get much more interesting than this for relational theories. The problem is that we cannot say anything about the operations in the theory, all we know is that either there is an operations of a certain arity or there is no such operation. We shall see throughout the thesis that when we take more complex theories into account, in particular ones in which we can specify the operations explicitly, we will get very interesting relationships between theories.

The category Rel is a dagger category. Simply put, this just means that every relations has a converse defined by $r^\circ(x, y) := r(y, x)$. Thus for every morphism between sets there are two canonical ways of making it into a relation. The first one is to consider its graph, and the second the converse of that graph.

So far, we have focused on the graph of $\iota : M \to A$. But if its converse, denoted $i^* : \tilde{A} \to \tilde{M}$, is also a monad morphism, we shall get, by Theorem 2.18, a 2-functor from $\mathsf{kl}(\tilde{A}) \to \mathsf{kl}(\tilde{M})$.

We directly check that ι^* is a natural transformation. Both passages in

$$\begin{tikzcd} AX \arrow[r, "\iota^*_X", "|"'] \arrow[d, "\tilde{A}r"', "|"] & MX \arrow[d, "\tilde{M}r", "|"'] \\ AY \arrow[r, "\iota^*_Y"', "|"] & MY \end{tikzcd}$$

.

relate a list l in AX to a list l' in MY precisely when the length of both lists is the same, l does not have any negative entries, and for all $x_i \in l,\ y_i \in l',\ \ x_i r y_i$.

ι^* is also a monad morphism. This too can be directly checked and follows an argument similar to the one above.

We thus get a 2-functor

$$J : \mathsf{kl}(\tilde{A}) \to \mathsf{kl}(\tilde{M});$$

this functor sends an A-theory $\psi : X \nrightarrow AX$ to the M-theory $X \xrightarrow{\psi} AX \xrightarrow{\iota^*_X} MX$. $J\psi([x_1, \ldots x_n], x) = 1$ provided that exist $k \geq 0$ and $y_1, \ldots, y_k \in X$ such that

$$\psi([-y_1, \ldots, -y_k, x_1, \ldots, x_n], x) = 1.$$

That is, the 2-functor J forgets the negative parts of the arities of each operation of ψ. It almost acts as a projection operator on each operations.

The two embeddings of Set into Rel create adjoints pairs

$$\iota_* \dashv \iota^*$$

in the later 2-category. Explicitly,

$$\iota_* \cdot \iota^* \subseteq (1_{AX})^* \text{ and } (1_{MX})^* \subseteq \iota^* \cdot \iota_*.$$

This relationship suggests that the 2-functors I and J are not entirely unrelated.

Let $F, G : \mathsf{Rel} \to \mathsf{Rel}$ be two 2-functors. We say that F is *locally left adjoint* to G provided that for every two sets X and Y there is an adjunction

$$\begin{tikzcd} \mathsf{Rel}(FX, Y) \arrow[r, shift left] \arrow[r, phantom, "\top"] & \mathsf{Rel}(X, GY) \arrow[l, shift left] \end{tikzcd}$$

Theorem 2.19. *J is locally left adjoint to I.*

Proof. Since both functors are identity on objects, we just need to show that there is an adjunction

$$I_{X,Y} \dashv J_{X,Y} : \mathsf{kl}(\tilde{A})(X, Y) \to \mathsf{kl}(\tilde{M})(X, Y).$$

This amounts to showing that for any $r : X \nrightarrow AY$ and $s : X \nrightarrow MY$ we have

$$Is \subseteq r \iff s \subseteq Jr.$$

or, equivalently,

$$i_* \cdot s \subseteq r \iff s \subseteq i^* \cdot r$$

For $\implies$,

$$s = 1 \cdot s \subseteq \iota^* \cdot \iota_* \cdot s \subseteq \iota^* \cdot r.$$

The converse is similar. $\square$

Thus a monad morphism $\alpha : S \to T$ that suitably interact with the distributive laws λ and ρ always gives a passage from theories of types S to theories of types T. Further, when α^* is a monad morphism, it also gives rise to a functor in the other direction. The two functors form an adjoint pair. In particular, when dealing with inclusions of monads, this adjoint pair can be interpreted as forgetting structure and putting free structure on a theory.

In the rest of this work we develop an analogue of the above theory for 2-categories. In particular, we examine 2-monads on Cat and their extensions to the category of profunctors. We shall see that internal monads in Kleisli bicategories of such extensions can describe many flavours of theories: plain operads, operads and Lawvere theories.

We then show that 2-monad morphisms that interact nicely with the 2-distributive laws can also be extended. These extensions allow us to build functors between the Kleisli bicategories where our theories live. We also show that, when certain conditions hold, the resulting functors form so-called lax 2-adjunctions.

Chapter 3

Categorical Universal Algebra

In Universal Algebra the formal specifications of a particular construct themselves become mathematical objects which can be investigated. Universal Algebra became an important part of modern mathematical landscape with the publication of *A Treatise on Universal Algebra* in 1898. Since then the field saw numerous advances and breakthroughs. One of those was the use of Category Theory as the formal mathematical system for describing specifications of various theories.

Peter May's theory of operads, initially introduced to tackle issues in algebraic topology, turned out to be very useful in specifying the semantics for higher dimensional categories [Leinster, 2003]. In parallel, Mc Lane and Adams developed PROs and PROPs which are special types of categories that contain all the data needed to describe algebraic gadgets in monoidal and symmetric monoidal categories. The field reached full maturity with Lawvere's discovery of finite product theories, or more colloquially, Lawvere theories. Like PROPs and PROs these are special types of categories which model algebraic specifications for some of the most commonly encountered algebraic structures. Their models live in finite-product categories.

In this chapter we begin with a general discussion of theories. We then give a brief introduction to operads. Following this, we explore PROPs and PROs. We conclude with Lawvere Theories. We also highlight the similarities and differences between all these tools.

3.1 Theories - an overview

For us a theory is a formal specification which consists of operations of certain *arities*, ways of *combining* these operations and a methods for specifying equations between the combined operations. For example, the specification for the theory of monoids involves operations of *linear* arities, e.g. operations that accept a finite list of inputs and produce a single output. Multiplications $m : X \times X \to X$ is one such operation; it accepts tuples of elements of X and returns a single element of X. The multiplicative identity $e : \{*\} \to X$ is an operations of arity 0. This multiplicative identity can be combined with the multiplication

$$X \simeq X \times \{*\} \xrightarrow{1 \times e} X \times X \xrightarrow{m} X$$

One equation in the theory of monoids says that this expression equals to the operation identity $\mathrm{id}_X : X \to X$.

Note that in the above specification we do not concern ourselves with what X is. That is, X here does not play the role of the carrier set for the monoid, but rather acts as a dummy variable that is used to specify the domain and codomain of operations.

Concrete monoids are called *models* of the above theory. Each model consists of the underlying set of the monoid and a concrete implementation for each operation in the specification such that the equations between the operations in the specification are also satisfied for those implementations.

Category Theory provides us with tools that allow for an elegant expression of both the specification of a theory and its category of models.The idea is to package the specifications of a theory into a category. The objects of this category play the role of X in the above specification, that is, they act as dummy variables. The morphisms of this category act as the operations of the theory. Equations are specified by equality between the morphisms in the category. Finally, the idea of arity is captured by adding some structure to this category. For example, to model arities of type list, we add monoidal structure to our category.

Given a category $\mathcal{G}$ which represents a particular theory, the models of this theory (in the category of sets and functions) are structure preserving functors

$$\mathcal{G} \to \mathsf{Set}.$$

This summarises the main ideas behind Categorical Universal Algebra. The implementation of these ideas is not trivial. First, there are several variants of structures for capturing algebra syntax. On one hand, there are operads. They model operations with multiple inputs and a single output. Operads themselves come in many flavours: plain, symmetric and coloured, to name a few. There are also versions of operads based on finite product structure.

There exist gadgets for capturing syntax of theories whose operations have multiple inputs and multiple outputs. They also come in several variants. The simplest are PROs, which are special kind of monoidal categories. PROPs are symmetric variant of PROs and Lawvere Theories are their finite-product counterpart.

3.2 Operads

The name operads first appeared in May's *The Geometry of Iterated Loop Spaces* []. May uses operads in the category of topological spaces as a tool for dealing with iterated loop spaces. But while May introduced the name, Operad appears as early as 1898 in Whitehead's *A Treatise on Universal Algebra*. They next reappear in Lazar's work in the 50's. By the late 60's Operads and PROPs were widely studied in the algebraic and topological communities, as witnessed by the 1967 seminar on PROPs organised by Mc Lane.

At these early stages much of the the focus was on symmetric and plain operads. Later developments in Categorical Universal Algebra, most notably the discovery of Lawvere theories, let to a greater variety of operads. Here we cover the most important variants, beginning with the simplest one: plain operads.

3.2.1 Plain operads

A *non-symmetric operad* or a *plain operad* consists of

- For each $n \in \mathbb{N}$, a set $A(n)$ of *operations of arity* n, which we visualise as
- The *identity operation* $\mathrm{id} \in A(1)$.

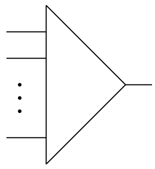


Diagram 3.1: An operation of arity n

- For $k, n_1, \ldots, n_k \in \mathbb{N}$ a *composition operation*

$$A(n) \times A(k_1) \times \ldots \times A(k_n) \xrightarrow{\kappa} A(k_1 + \ldots + k_n)$$

  which we visualise as

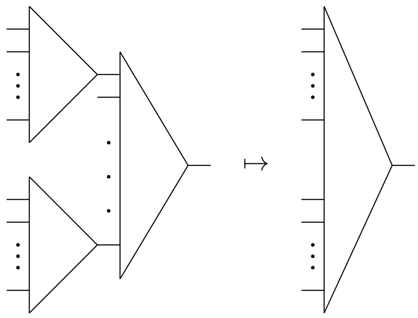


Diagram 3.2: Operad composition

- Composition is associative and the identity acts as the unit for the composition:

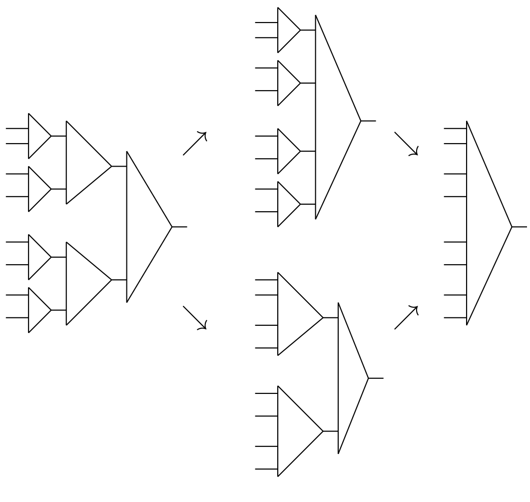


Diagram 3.3: Associativity of operad composition

Note that there are more general definitions of operads, where the elements of the sequence $A(k)$ are not sets, but objects in an arbitrary monoidal category. Indeed, May's original definition [May, 1972] took $A(k)$ to be topological spaces. In our approach, we will be content with taking $A(k)$ to be sets, so that, using standard terminology, our operads are operads in the monoidal category $(\mathsf{Set}, \times, \{*\})$.

Given two plain operads A and B, a *morphism of plain operads* consists of

- for each $n \in \mathbb{N}$ a function $f_n : A(n) \to B(n)$;

- such that $f_1(\mathsf{id}) = \mathsf{id} \in B(1)$
- and composition is preserved:

$$\begin{tikzcd}
A(k) \times A(n_1) \times \ldots \times A(n_k) \arrow[r, "\kappa"] \arrow[d, "f_k \times f_{n_1} \times \ldots \times f_{n_k}"'] & A(n_1 + \ldots + n_k) \arrow[d, "f_{\sum n_k}"] \\
B(k) \times B(n_1) \times \ldots \times B(n_k) \arrow[r, "\kappa"] & B(n_1 + \ldots + n_k)
\end{tikzcd}$$

Plain operads and their morphisms form a category.

Example 3.1. The simplest plain operad E is given by

$$E(1) = \{*\} \text{ and } E(n) = \emptyset, \ \ \forall n \neq 1.$$

It provides specifications for the theory that does not contain any non-trivial operations.

Example 3.2. Define an operad S by $S(n) = \{*\}$, $\forall n \geq 1$ and $S(0) = \emptyset$. Since the singleton is terminal in Set, composition and the unit are defined in the obvious way. S is called the operad for semi-groups.

Example 3.3. Any object x in a monoidal category $\mathcal{C}$ gives rise to a plain operad $\mathsf{End}(x)$ by setting

$$\mathsf{End}(x)(n) = \mathcal{C}(x^{\otimes n}, x).$$

Composition

$$\mathsf{End}(x)(k) \times \mathsf{End}(x)(n_1) \times \ldots \times \mathsf{End}(x)(n_k) \to \mathsf{End}(x)(n_1 + \ldots + n_k)$$

is given by

$$(f, g_1, \ldots, g_k) \mapsto f \cdot (g_1 \otimes \ldots \otimes g_k).$$

This operad is called the *endomorphism operad of x*.

Let $\mathcal{C}$ be a monoidal category and $x \in \mathrm{ob}\mathcal{C}$ be an object. Let A be an operad. An *A-algebra structure on x* is a morphism of operads

$$A \to \mathsf{End}(x).$$

Example 3.4. A semi-group structure on a set X is equivalent to a operad morphism $s : S \to \mathsf{End}(X)$. Indeed, such an operad morphism picks a function

$$m := s(2)(*) \in \mathsf{Set}(X \times X, X).$$

that acts as the multiplication of the semi-group since

$$m \cdot (1 \times m) = m \cdot (m \times 1) \tag{3.1}$$

Indeed, the left-hand-side is the image of

$$\mathsf{End}(X)(2) \times \big(\mathsf{End}(X)(1) \times \mathsf{End}(X)(2)\big) \xrightarrow{\kappa} \mathsf{End}(X)(3),$$

while the right-hand-side is the image of

$$\mathsf{End}(X)(2) \times \big(\mathsf{End}(X)(2) \times \mathsf{End}(X)(1)\big) \xrightarrow{\kappa} \mathsf{End}(X)(3).$$

Since s is a morphism of operads, the following diagram commutes

$$\begin{tikzcd}
S(2) \times \big(S(1) \times S(2)\big) \arrow[r, "\kappa"] \arrow[d, "s"'] & S(3) \arrow[d, "s_3"] & S(2) \times \big(S(2) \times S(1)\big) \arrow[l, "\kappa"'] \arrow[d, "s"] \\
\mathsf{End}(X)(2) \times \big(\mathsf{End}(X)(1) \times \mathsf{End}(X)(2)\big) \arrow[r, "\kappa"'] & \mathsf{End}(X)(3) & \mathsf{End}(X)(2) \times \big(\mathsf{End}(X)(2) \times \mathsf{End}(X)(1)\big) \arrow[l, "\kappa"]
\end{tikzcd}$$

showing that both sides of Equation 3.1 equal $s_3 \cdot \kappa(*)$.

Example 3.5. For a plain operad A and an object x of a monoidal category $\mathcal{C}$, the *Schur functor*

$$S_A(x) := \coprod_{n \geq 1} A(n) \otimes x^{\otimes n}$$

gives the formula for an A-algebra on x. See [Markl et al., 2007] for a detailed treatment.

Taking $\mathcal{C} = (\mathsf{Set}, \times, \{*\})$, a typical element of $S_A(X)$ is

$$(\varphi, x_1, \ldots, x_n)$$

with $\varphi \in A(n)$ and $x_i \in X$. But this is just another way of formally writing down $\varphi(x_1, \ldots, x_n)$.

The Schur functor gives the free A-algebra on x.

3.2.2 Symmetric operads

A *symmetric operad* consists of a plain operad A equipped, for each $n \in \mathbb{N}$, with right actions of symmetric groups

$$\gamma_n : A(n) \times S_n \to A(n).$$

These actions specify the interaction of each operation with symmetry. We visualise the actions as follows

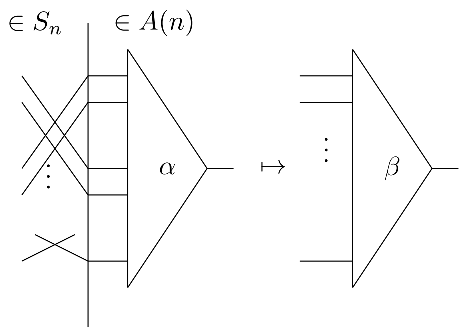
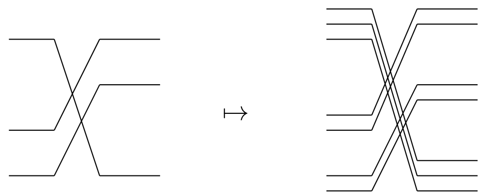


Diagram 3.4: Action of S_n on the operations in $A(n)$

Composition needs to be compatible with the actions of the symmetric groups. This condition is called *equivariance* and it is not entirely trivial. Let $n_1, \ldots, n_k$ be a list of natural numbers that add-up to m. Any such list induces a mapping

$$s_{n_1,\ldots,n_k} : S_k \to S_{\sum_{i=1}^k n_i} = S_m, \quad \bar{\sigma} := s_{n_1,\ldots,n_k}(\sigma) \tag{3.2}$$

where $\bar{\sigma}$ permutes the list $n_1, \ldots, n_k$ by permuting the indices. This is best illustrated with an example: $s_{n_1,\ldots,n_k}$ allows us to define the passage



σ $\qquad\qquad$ $\bar{\sigma}$

Diagram 3.5: Example application of $s_{n_1,\ldots,n_k}$ to σ

$$\Sigma = \Sigma_{(\sigma,n_1,\ldots,n_k)} : S_k \times \prod_{i=1}^{k} A(n_i) \to \prod_{i=1}^{k} A(n_{\sigma(i)}) \times S_m \tag{3.3}$$

by sending

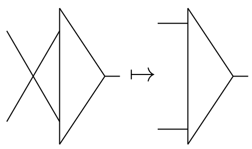
$$\big(\sigma, (x_1, \ldots, x_k)\big) \mapsto \big((x_{\sigma(1)}, \ldots, x_{\sigma(k)}), \bar{\sigma}\big)$$

We say that *equivariance holds for* A provided that the following diagram commutes for all k, $n_1, \ldots, n_k$, $m = \sum_{i=0}^k n_i$

$$\begin{tikzcd}
A(k) \times S_k \times \prod_{i=1}^k A(n_i) \arrow[r, "1\times\Sigma"] \arrow[dd, "\gamma_k\times 1"'] & A(n) \times \prod_{i=1}^k A(n_i) \times S_m \arrow[d, "\kappa\times 1"] \\
& A(m) \times S_m \arrow[d, "\gamma_m"] \\
A(k) \times \prod_{i=1}^k A(n_i) \arrow[r, "\kappa"'] & A(m)
\end{tikzcd} \tag{3.4}$$

This is visualised in Diagram 3.6.

Example 3.6. Define an operad C whose underlying plain operad is the operad S of Example 3.2. The actions of the symmetric groups are trivial, which gives the commutativity of operations:



We call this operad *the operad for commutative semi-groups.*

Example 3.7. Any object x in a symmetric monoidal category $\mathcal{C}$ gives rise to an operad $\mathsf{End}(x)$ by setting

$$\mathsf{End}(x)(n) = \mathcal{C}(x^{\otimes n}, x).$$

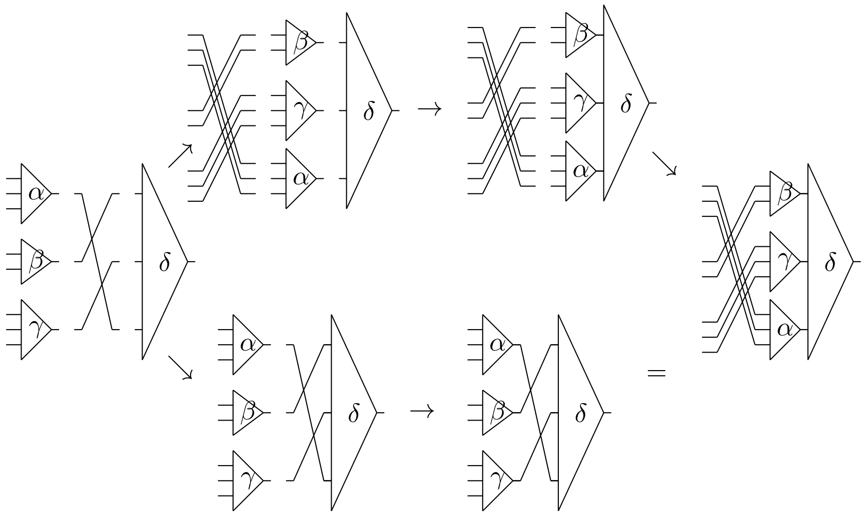


Diagram 3.6: Equivariance

The composition and unit are the same as for the plain case (cf. Example 3.3). Actions of symmetric groups

$$\mathcal{C}(x^{\otimes n}, x) \times S_n \to \mathcal{C}(x^{\otimes n}, x)$$

are defined by identifying each $\sigma \in S_n$ with the corresponding symmetry isomorphism $\sigma' : x^{\otimes n} \to x^{\otimes n}$ in $\mathcal{C}$ and sending the pair (σ, φ) to the composite $\varphi \cdot \sigma'$.

A *morphism of symmetric operads*, $\alpha : A \to B$ is a morphism of the underlying plain operads with the additional property that the actions of the symmetric groups are preserved, that is for each $n \in \mathbb{N}$,

$$\begin{tikzcd} A(n) \times S_n \arrow[r, "\gamma_n"] \arrow[d, "\alpha_n \times 1"'] & A(n) \arrow[d, "\alpha_n"] \\ B(n) \times S_n \arrow[r, "\gamma'_n"'] & B(n) \end{tikzcd}$$

commutes.

Let x be an object in a symmetric monoidal category $\mathcal{C}$ and let A be an operad. An A-algebra on x is a morphism of symmetric operads

$$A \to \mathsf{End}(x)$$

Example 3.8. Giving a set X the structure of a commutative semi-group is exactly the same as giving a morphism of operads $C \to \mathsf{End}(X)$. Given such a morphism $c : C \to \mathsf{End}(X)$, we obtain an associative multiplication as in Example 3.4. m is commutative because the square

$$\begin{tikzcd} C(2) \times S_2 \arrow[r, "\gamma_2"] \arrow[d, "c_2 \times 1"'] & C(2) \arrow[d, "c_2"] \\ \mathsf{End}(X)(2) \times S_2 \arrow[r, "\gamma'_2"'] & \mathsf{End}(X)(2) \end{tikzcd}$$

commutes. Indeed, the bottom arrow sends (m, σ) to the composite $(m \cdot \sigma)$ in Set and this composite equals $m = c_2(*)$.

3.2.3 Coloured operads

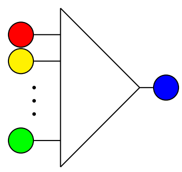
So far we have focused on operads where the inputs and outputs of every operations were of the same type. There are instances when we wish to distinguish the types of inputs and outputs from each other. The simplest examples arise in computing when functions can input different types of arguments (strings, integers, floats, etc) and output other types.

Coloured operads provide semantics for describing such operations. A *plain coloured operad* or a *multicategory* consists of the following ingredients

- A class X of objects (also called colours);
- for each $x \in X$, and a list $[x_1, \ldots, x_n]$ of X objects, a set
$$A([x_1, \ldots, x_n], x)$$
of operations from $[x_1, \ldots, x_n]$ to x;
- for each $x \in X$ an element id $\in A([x], x)$;
- for each list $[x_1, \ldots, x_n]$, $x \in X$ and lists $[y_{11}, \ldots, y_{1k_1}], \ldots [y_{n1}, \ldots, y_{nk_n}]$ a composition operation
$$A([x_1, \ldots, x_n], x) \times \big(A([y_{11}, \ldots, y_{1k_1}], x_1) \times \ldots \times A([y_{n1}, \ldots, y_{nk_n}], x_n)\big) \to A([y_{11}, \ldots, y_{nk_n}], x)$$
such that composition is associative and is compatible with the unit.

Adding actions of symmetries, as was done Section 3.2.2 for plain operads, produces *symmetric coloured operads*, or *symmetric multicategories*.

We visualise operations of coloured operads as follows.



A coloured operad with a single colour is just a regular operad.

3.2.4 Other types of operads

While symmetric operads are the most studied type, important variants do exist. In *braided operads*, for example, the actions of symmetric groups S_n are replaced by the action of braid groups B_n.

We can also replace groups by homsets. So, for example, the action of the symmetric group S_n can be replaced
$$A(n) \times \mathbb{P}(n, n) \to A(n)$$
where $\mathbb{P}$ is the skeletal category of finite sets and bijections. In this setting, the axioms of groups actions are equivalent to compatibility with composition and the identity. More generally, let $\mathcal{A}$ be a monoidal category

whose objects are the natural numbers. Then an *operad of type* $\mathcal{A}$ -or an $\mathcal{A}$-operad - is an operad A equipped, for each $n, k \in \mathbb{N}$, with a function

$$\alpha_{n,k} : A(k) \times \mathcal{A}(n,k) \to A(n)$$

such that for all $n, k, m \in \mathbb{N}$, the following diagrams commute

$$\begin{tikzcd}
A(k) \times \mathcal{A}(n,k) \times \mathcal{A}(m,n) \arrow[r, "1\times\circ"] \arrow[d, "\alpha\times 1"'] & A(k) \times \mathcal{A}(m,k) \arrow[d, "\alpha"] \\
A(n) \times \mathcal{A}(m,n) \arrow[r, "\alpha"'] & A(m)
\end{tikzcd}
\qquad
\begin{tikzcd}
A(k) \arrow[r, "1\times\mathrm{id}"] \arrow[dr, "1"'] & A(k) \times \mathcal{A}(k,k) \arrow[d, "\alpha"] \\
& A(k),
\end{tikzcd}$$

In order to define equivariance, we also need a passage

$$\Gamma : \mathcal{A}(n,k) \times \prod_{i=1}^{n} A(k_i) \to \prod_{i=1}^{n} A(k_i) \times \mathcal{A}(\sum_{i=1}^{n} k_i, m') \tag{3.5}$$

for some m'. We can then define equivariance as we have done in diagram 3.4. Although, in practice it is rather difficult to obtain such a passage.

Example 3.9. It turns out that when $\mathcal{A} = \mathbb{F}^{\text{op}}$ is the opposite of the skeletal category of finite sets, there is a sensible way to define the mapping in 3.5. Note that $\mathbb{F}^{\text{op}}$ consist of symmetries, copying operations and delete operations. We define Γ on symmetries to equal Σ of 3.3. On deletion operations Γ is given by

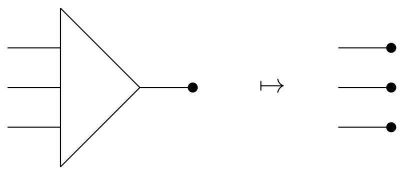


Diagram 3.7: The action of Γ on deletions

and on copying operations it is given by

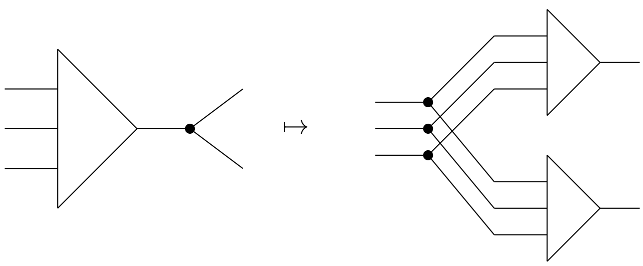


Diagram 3.8: The action of Γ on copy operations

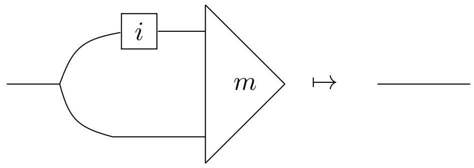
So Γ copies and deleted operations of A-operations.

A *Cartesian operad* is an operad of type $\mathbb{F}^{\text{op}}$. Since $\mathbb{F}^{\text{op}}$ is the free finite-product category on 1, the actions

$$A(k) \times \mathbb{F}^{\text{op}}(n,k) \to A(n)$$

specify how the operations in A interact with finite product structure.

Since finite product structure allows us to copy and delete inputs, Cartesian operads allows us to specify the axioms for the theory of groups. Let G be the Cartesian operad for groups. Then the fact that every element has an inverse is captured by the equation



(here $i : 1 \to 1$ is the inverse operation and $m : 2 \to 1$ is the multiplication.)

We treat Cartesian operads formally in Section 6.2.3 of Chapter 6. For now, we hope that this brief introduction conveyed both the importance of Cartesian operads and the difficultly in with attempting to define them naively.

3.3 PROs, PROPs and PROBs

Operads model theories with multiple inputs and a single output. This constraint means that specifications for several important algebraic structures cannot encoded by an operad. A bialgebra is a classic example of such a structure. Bialgebras are equipped with both a multiplication and a comultiplication, and the latter cannot be expressed in an operad.

PROPs and their variants are a generalisation of operads which can express algebraic operations with multiple inputs and multiple outputs. In this section we study PROs, PROPs and Lawvere theories. These correspond to plain, symmetric and Cartesian operads.

3.3.1 PROs

A *PROduct category* or simply a *PRO* is a strict monoidal category $\mathbb{T} = (\mathbb{T}, \otimes, I)$ with $\mathrm{ob}\mathbb{T} = \mathbb{N}$, $\otimes = +$ and $I = 0$. The free monoidal category on a single object is a PRO; its objects are the natural numbers and the only morphisms are the equalities. We denote this category by $\mathbb{N}$. The simplicial category, Δ, (see [**?**]) is a classic example of a PRO. For each $n \in \mathrm{ob}\Delta$, write

$$[n] = \{0 < 1 < \ldots < n-1\}$$

and set

$$\Delta(n, m) = \{f : [n] \to [m] \mid f \text{ monotone } \}.$$

Δ has two subcategories Δ_m and Δ_e whose morphisms are the monomorphisms in Δ and the epimorphisms in Δ, respectively. These subcategories are themselves PROs.

Any object A is a monoidal category $\mathcal{C}$ induces a PRO $\langle A, A\rangle$; the hom-sets are given by

$$\mathsf{End}(A)(n, m) = \mathcal{C}(A^{\otimes n}, A^{\otimes m}).$$

Composition and the tensor are inherited from $\mathcal{C}$.

A *morphism of PROs* from $\mathbb{T}$ to $\mathbb{S}$ is a strict monoidal identity-on-objects functor $H : \mathbb{T} \to \mathbb{S}$.

The main interest in PROs stems from their ability to describe algebraic gadgets. An *algebra* of a PRO $\mathbb{T}$ in a monoidal category $\mathcal{C}$, or a $\mathbb{T}$-*algebra* in $\mathcal{C}$, is a strict monoidal functor

$$F : \mathbb{T} \to \mathcal{C}.$$

Since F is strict monoidal, its image on $\mathrm{ob}\mathbb{T}$ is completely determined by $F1$. For a given object $A \in \mathrm{ob}\mathcal{C}$ a $\mathbb{T}$-*algebras structure on* A is an $\mathbb{T}$-algebra F with $F1 = A$.

Alternatively, we can define *an* $\mathbb{T}$-*algebra structure on an object* A to be a morphism of PROs

$$\mathbb{T} \to \mathsf{End}(A).$$

Given two algebras F and G a *morphism of* $\mathbb{T}$-*algebras* is a monoidal natural transformation

$$\alpha : F \to G.$$

Such α are completely determined by their component at 1: $\alpha_n = \alpha_{1+\ldots+1} = \alpha_1 \otimes \ldots \otimes \alpha_1$.

$\mathbb{T}$-algebras in $\mathcal{C}$ and $\mathbb{T}$-algebra morphisms form a category $\mathbb{T}$-Alg.

Example 3.10. $\mathbb{N}$-algebras in $\mathcal{C}$ are just the objects of $\mathcal{C}$. This is not surprising because $\mathbb{N}$ does not contain any information - it has no non-trivial morphisms. $\mathbb{N}$-algebra morphisms are then just the morphisms of $\mathcal{C}$ and so $\mathbb{N}$-$\mathsf{Alg} \simeq \mathcal{C}$.

Example 3.11. Δ-algebras are more interesting. Since $1 \in \Delta$ is the terminal object, there are unique morphisms

$$m : 2 \to 1 \text{ and } u : 0 \to 1$$

in Δ, and

$$\begin{tikzcd} 3 \arrow[r, "m+1"] \arrow[d, "1+m"'] & 2 \arrow[d, "m"] \\ 2 \arrow[r, "m"'] & 1 \end{tikzcd} \qquad \begin{tikzcd} 1 \arrow[r, "1+u"] \arrow[dr, "1"'] & 2 \arrow[d, "m"] & 1 \arrow[l, "u+1"'] \arrow[dl, "1"] \\ & 1 & \end{tikzcd} \tag{3.6}$$

commute; thus $(1, m, u)$ is a monoid in Δ. A Δ-algebra $F : \Delta \to \mathcal{C}$ preserves the above diagrams and hence Δ-algebras give rise to monoids $(A := F1, \mu := Fm, \eta := Fu)$ in $\mathcal{C}$. Conversely, a given monoid (A, μ, η) in $\mathcal{C}$ defines a Δ-algebra F_A. Obviously $F_A 1 := A$, $F_A m := \mu$ and $F_A u := \eta$. In fact, these three assignments determine F_A on all of Δ. Indeed, since F_A is strict monoidal, it is fully determined on $\mathrm{ob}\Delta$. Let $m^k : k \to 1$ denote the only such morphism $(m^0 = u, m^1 = 1, m^2 = m)$. Since 1 is terminal

$$m^k = m \cdot (m^{k-1} + 1).$$

for any $k > 0$. Thus, the definition of F_A extends to all morphisms of the form m^k by induction. Since any $f : n \to r$ in Δ can be written as

$$f = m^{|f^{-1}(0)|} + \ldots + m^{|f^{-1}(r-1)|}.$$

F_A extends to all of Δ. Monoid morphisms $f : A \to B$ in $\mathcal{C}$ give rise to monoidal natural transformation

$F_f : F_A \to F_B$ defined by

$$(F_f)_n := f^{\otimes n}.$$

And monoidal natural transformation $\alpha : F \to G$ defines monoid morphisms $\alpha_1 : F1 \to G1$ in $\mathcal{C}$. Indeed, preservation of multiplication follows from naturality

$$\begin{tikzcd} F2 \arrow[r, "\alpha_1+\alpha_1"] \arrow[d, "m"'] & G2 \arrow[d, "m"] \\ F1 \arrow[r, "\alpha_1"'] & G1. \end{tikzcd}$$

In fact there is a functor

$$F_{(-)} : \mathsf{Mon}(\mathcal{C}) \to \Delta\text{-}\mathsf{Alg}$$

from the category of monoids in $\mathcal{C}$ and their homomorphisms to the category of Δ-algebras and their morphisms sending monoids to Δ-algebras and a functor

$$(-)1 : \Delta\text{-}\mathsf{Alg} \to \mathsf{Mon}(\mathcal{C})$$

sending Δ-algebras to monoids. These functors form an isomorphism of categories. Writing $\mathsf{Cat}(1, \mathsf{Mon}(\mathcal{C}))$ for $\mathsf{Mon}(\mathcal{C})$ allows for a succinct expression of this fact:

$$\mathsf{MonCat}(\Delta, \mathcal{C}) \simeq \mathsf{Cat}(1, \mathsf{Mon}(\mathcal{C})).$$

From this perspective, Δ can be thought of as *the free strict monoidal category containing a monoid object.*

Example 3.12. Δ^{op}-algebras in $\mathcal{C}$ are precisely the comonoids in $\mathcal{C}$. Δ^{op}-algebra homomorphisms in $\mathcal{C}$ are the comonoid homomorphisms in $\mathcal{C}$. The details are dual to those in Example 3.11.

Example 3.13. Using the techniques developed in (2), one can show that Δ_e-algebras are the semigroups in $\mathcal{C}$ and the Δ_m-algebras are the I-pointed objects in $\mathcal{C}$. (A is an *I-pointed object* provided that it is equipped with a morphism $\alpha : I \to A$).

3.3.2 PROPs

A *PROduct and Permutation category* or *PROP* is a symmetric monoidal category $\mathbb{T} = (\mathbb{T}, \otimes, I)$ with $\mathrm{ob}\mathbb{T} = \mathbb{N}$, $\otimes = +$ and $I = 0$. Thus a PROP is nothing but a symmetric PRO. The simplest PROP is one without any structure, that is, just the free symmetric monoidal category on a single object category; we denote it by $\mathbb{P}$. Explicitly, for any $n, m \in \mathbb{N}$,

$$\mathbb{P}(n, m) = \begin{cases} S(n), & \text{if } n = m; \\ \emptyset, & \text{else,} \end{cases}$$

($S(n)$ is the symmetric group with n elements).

$\mathbb{P}$ is certainly a PRO. A $\mathbb{P}$-algebra $\tilde{A} : \mathbb{P} \to \mathcal{C}$ in a monoidal category $\mathcal{C}$ assigns the non-trivial permutation $1+1 \to 1+1$ to some morphism $A \otimes A \to A \otimes A$ that we shall call the symmetry isomorphism of A. If we let $\mathcal{C} = \mathbb{T}$ be a PRO, we obtain the following result:

Lemma 3.14. *A PROP $\mathbb{T}$ is a PRO $\mathbb{T}$ with a given strict monoidal identity-on-objects functor*

$$J : \mathbb{P} \to \mathbb{T}.$$

The skeletal category of finite sets and functions, $\mathbb{F}$, is also a PROP.

PROPs are important because, through their algebras, they describe algebraic gadgets in the symmetric monoidal setting. An *algebra of a PROP $\mathbb{T}$ in a symmetric monoidal category $\mathcal{C}$*, or simply a *$\mathbb{T}$-algebra in $\mathcal{C}$*, is a strict monoidal functor

$$F : \mathbb{T} \to \mathcal{C}.$$

Given two such algebras, a *morphism of $\mathbb{T}$-algebras* is a monoidal natural transformation $\alpha : F \to G$. We denote the category of $\mathbb{T}$-algebras and their morphisms by $\mathbb{T}$-Alg.

Example 3.15. The category of $\mathbb{P}$-algebras in $\mathcal{C}$ is just $\mathcal{C}$ itself.

Example 3.16. The "combinatorial" structure of $\mathbb{F}$ resembles that of Δ. The fact that 1 is terminal gives rise to $m : 2 \to 1$ and $u : 0 \to 1$. The diagrams of equation (3.6) still commute. But $\mathbb{F}$ has more morphisms than Δ – there are two morphisms from $2 \to 2$. We call the non-identity one $s : 2 \to 2$, and this morphism interacts with the multiplication via

$$\begin{tikzcd} 2 \arrow[rr, "s"] \arrow[dr, "m"'] & & 2 \arrow[dl, "m"] \\ & 1 & \end{tikzcd}$$

The commutative diagrams of (3.6) along with the diagram above show that $(1, m, u)$ is a *commutative* monoid in $\mathbb{F}$. Hence any $\mathbb{F}$-algebra G gives rise to a commutative monoid $(G1, Gm, Gu)$ in $\mathcal{C}$. As in (2) of Examples 3.10, there is an isomorphism

$$\mathsf{SymMonCat}(\mathbb{F}, \mathcal{C}) \simeq \mathsf{Cat}(1, \mathsf{CommMon}(\mathcal{C})), \tag{3.7}$$

where $\mathsf{SymMonCat}$ is the 2-category of symmetric monoidal categories and $\mathsf{CommMon}(\mathcal{C})$ is the category of all the commutative monoids in $\mathcal{C}$ and monoid homomorphisms between them. Equation (3.7) shows that $\mathbb{F}$ plays a role very similar to Δ - it is the free symmetric monoidal category on 1 that contains a commutative monoid.

Example 3.17. $\mathbb{F}^{\mathrm{op}}$-algebras in $\mathcal{C}$ are precisely the cocommutative comonoids and their homomorphisms.

3.3.3 PROs vs. Operads

PROs, PROPs and Operads are not unrelated. Every plain operad gives rise to a PRO, but there are PROs that are not induced by plain operads. There are analogous results for PROPs and symmetric operads.

Let S be an operad. We build a PRO S' by setting

$$S'(n,1) = S(n)$$

and building operations in $S'(n,k)$ by tensoring and composing morphisms from $S'(r,1)$ for various r. We can similarly construct a morphisms of PROs from a morphisms of operads. This gives a functor

$$I : \mathsf{POp} \to \mathsf{PRO}$$

from the category of plain operads to the category of PROs. Similarly, each symmetric operad gives rise to a PROP. So we also have

$$J : \mathsf{Op} \to \mathsf{PROP}$$

Since every PROP is a PRO and every operad is a plain operad, we end up with the following commutative square:

$$\begin{tikzcd} \mathsf{POp} \arrow[r, "I"] & \mathsf{PRO} \\ \mathsf{Op} \arrow[u, hook] \arrow[r, "J"'] & \mathsf{PROP} \arrow[u, hook] \end{tikzcd}$$

An example of a PRO that is not given by an operad is the PRO for comonoids. The comultiplication $\delta : 1 \to 2$ cannot be represented by an operad.

3.4 Lawvere Theories

Lawvere theories can be thought of as Cartesian PROPs. They model an extensive class of algebraic structures and have many fascinating properties. As a result, they are the most studied of all structures in Categorical Universal Algebra.

3.4.1 Definitions and examples

A *Lawvere Theory* is a category L with a countable set of objects $\{A^0, A^1, A^2, \ldots\}$ such that $A^n \simeq (A^1)^n$, i.e. each A^n is the n'th power of A^1.

Example 3.18. The simplest Lawvere theory is the free finite product category on the one-object category $\mathbf{1}$, $\mathbb{F}^{\mathrm{op}}$. category of finite sets and all morphisms between them.

Any Lawvere theory will contain $\mathbb{F}^{\mathrm{op}}$ as a subcategory. In fact, we have

Proposition 3.19. *There is a one-to-one correspondence between Lawvere theories Ł and identity-on-object finite-product-preserving functors* $\mathbb{F}^{\mathrm{op}} \to \mathcal{C}$.

A *model* or an *algebra* of a Lawvere theory L in a finite product category $\mathcal{C}$ is a product preserving functor $F : L \to \mathcal{C}$. A *morphism of models* is a natural transformation between such functors. This data arranges itself naturally into a category $\mathsf{Mod}(L, \mathcal{C})$.

Example 3.20. Let $\mathcal{C}$ be a category with finite products. A finite product preserving functor $X : \mathbb{F}^{\mathrm{op}} \to \mathcal{C}$ gives a choice of an object $x := X1$ in $\mathcal{C}$. And any object in $x \in \mathcal{C}$ gives rise to such a functor. Set $X0 = \top$, the terminal object, $X1 = x$ and $Xn = x^n$. On morphisms the projections in $\mathbb{F}^{\mathrm{op}}$ are mapped to projections in $\mathcal{C}$. A morphism of models $\alpha : X \to Y$ in $\mathsf{Mod}(\mathbb{F}^{op}, \mathcal{C})$ gives a morphism $\alpha_1 : x \to y$ in $\mathcal{C}$. Any morphism $f : x \to y$ gives rise to a natural transformation $f : X \to Y$ by setting $f_n := f^n$. Thus we proved

$$\mathsf{Mod}(\mathbb{F}^{\mathrm{op}}, \mathcal{C}) \simeq \mathcal{C}.$$

Example 3.21. Let $\mathbb{M}$ be the full subcategory of Mon with objects all free monoids on finite sets. Then $\mathbb{M}^{op}$ is the Lawvere theory for monoids: $\mathsf{Mod}(\mathbb{M}^{\mathrm{op}}, \mathsf{Set}) \simeq \mathsf{Mon}$. To see that $\mathbb{M}$ is indeed is a Lawvere theory, we need to exhibit an indentity-on-objects functor $\mathbb{F} \to \mathbb{M}$. But $\mathbb{F}$ can be viewed as a subcategory of $\mathbb{M}$, namely $\mathbb{F} \simeq F(\mathbb{F}) \hookrightarrow \mathbb{M}$, with F the restriction of the free monoid functor $F : \mathsf{Set} \to \mathsf{Mon}$.

Let $X : \mathbb{M}^{\mathrm{op}} \to \mathsf{Set}$ be a product preserving functor. We write n for the free monoid on n generators. Then X gives a set $x := X1$. In $\mathbb{M}$ there is a morphism $k : 1 \to 2$ which is determined by sending the generator $*$ of 1 to the sum of generators of $2 = \langle a, b \rangle$. That is

$$k(*) = a + b$$

with a, b the generators of 2. We also note the existence of the constant map $v : 1 \to 0$. One can now check that the following diagrams in $\mathbb{M}$ commute in $\mathbb{F}$:

$$\begin{tikzcd}
1 & 2 \arrow[l, "v+1"'] \arrow[r, "1+v"] & 1 \\
& 1 \arrow[u, "k"] \arrow[ul, equal] \arrow[ur, equal] &
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
1 \arrow[r, "k"] \arrow[d, "k"'] & 2 \arrow[d, "k+1"] \\
2 \arrow[r, "1+k"'] & 3
\end{tikzcd}$$

For example, in the rectangle the passages send $*$ to $(x + y) + z$ and $x + (y + z)$ which are equal in 3. Thus $(1, k, v)$ is a comonoid in $\mathbb{M}$ and hence a monoid $(1, m = k^{\mathrm{op}}, u = v^{\mathrm{op}})$ in $\mathbb{M}^{op}$. The functoriality of X, along with the fact that it preserves finite products, makes $(x, X(m), X(u))$ a monoid in Set.

Conversely, given a monoid (x, μ, η) (in Set) one defines a functor $X : \mathbb{M}^{\mathrm{op}} \to \mathsf{Set}$ by setting $Xn = x^n$. There are two types of morphism in $\mathbb{M}$, injections, which send generators to generators, and structure morphisms, like m above, which exploit the fact that generators can be combined using the monoid structure of each n. Each injection in $\mathbb{M}$ corresponds to a projection in the finite product structure of $\mathbb{M}^{\mathrm{op}}$. X sends these to the corresponding projections $X^n \to X$ in Set. For each n there is exactly one morphism $m_n : n \to 1$ in $\mathbb{M}^{op}$, where $m_2 = m$. We define

$$Xm_0 = \eta, \quad Xm_2 = \mu, \quad Xm_n = \mu \cdot (Xm_{n-1} \times 1_x)$$

Associativity and unit laws of (x, μ, η) guarantee that X is well defined. This completely determines X on the rest of the morphism of $\mathbb{M}^{\mathrm{op}}$ since these are all built from the $m_n's$ and the finite product structure.

Note that Set can be replaced by any category with finite products. Then the same argument shows that $\mathsf{Mod}(\mathbb{M}^{\mathrm{op}}, \mathcal{C}) \simeq \mathsf{Mon}(\mathcal{C})$, where $\mathsf{Mon}(\mathcal{C})$ is the category of internal monoids in the monoidal category

$(\mathcal{C}, \times, \top)$.

A *morphism of Lawvere theories* or an *algebraic functor* is a identity-on-objects finite-product-preserving functor $F : L \to L'$. For example, the inclusion $\mathbb{F}^{\mathrm{op}} \hookrightarrow \mathbb{M}^{\mathrm{op}}$ is a morphism of Lawvere theories. Lawvere theories and morphisms between them form the category Law.

For a fixed category $\mathcal{C}$ with finite products, there is a functor

$$\mathsf{Mod}(-, \mathcal{C}) : \mathsf{Law}^{\mathrm{op}} \to \mathsf{Cat}$$

where, for an algebraic functor $F : L \to L'$ and and L'-model $G : L' \to \mathcal{C}$ one sets $\mathsf{Mod}(F, \mathcal{C})(G) = G \cdot F$. The inclusion $\mathbb{F}^{\mathrm{op}} \hookrightarrow L$ gives rise to

$$U : \mathsf{Mod}(L, \mathcal{C}) \to \mathsf{Mod}(\mathbb{F}^{\mathrm{op}}, \mathcal{C}) \simeq \mathcal{C}.$$

When $\mathcal{C} = \mathsf{Set}$, U sends each model $X : L \to \mathsf{Set}$ to its underlying set, $X1$. We call U the *forgetful functor*.

When the category $\mathcal{C}$ is sufficiently nice, in particular cocomplete, U has a left-adjoint $F : \mathcal{C} \to \mathsf{Mod}(L, \mathcal{C})$. We record this below:

Theorem 3.22. *Let L be a Lawvere theory. Then the forgetful functor $U : \mathsf{Mod}(L, \mathcal{C}) \to \mathcal{C}$ has a left adjoint $F : \mathcal{C} \to \mathsf{Mod}(L, \mathcal{C})$. F is computed as the left Kan extension of the functor $X : \mathbb{F}^{\mathrm{op}} \to \mathcal{C},\ \ 1 \mapsto X$ along the inclusion $\mathbb{F}^{\mathrm{op}} \hookrightarrow L$.*

Example 3.23. The forgetful functor $\mathsf{Mod}(\mathbb{M}^{\mathrm{op}}, \mathsf{Set}) \simeq \mathsf{Mod} \to \mathsf{Set}$ has a left adjoint $F : \mathsf{Set} \to \mathsf{Mon}$. The underlying set of the free monoid on a set X is given by the formula

$$\begin{aligned} FX(1) &= \int^{m \in \mathbb{F}^{\mathrm{op}}} X(m) \times \mathbb{M}^{op}(1, m) \\ &\simeq \coprod_{m \in \mathbb{F}^{\mathrm{op}}} X^m \times \mathbb{M}^{\mathrm{op}}(1, m) / \sim \end{aligned}$$

Comparing the above formula to the equation for the Schur functor of Example 3.5, we see that the free model of a Lawvere theory and the free algebra for an operad are based on the same underlying idea.

3.4.2 Combining Lawvere Theories

Since a Lawvere theory is a special kind of category, we can combine two of them by combining the underlying categories. One natural method for combining theories, is by studying limits and colimits in the category Law. It turns out that the category Law is complete and cocomplete. This allows us to define the *sum* of Lawvere theories L and L' to be their coproduct in the category Law. One thinks about $L + L'$ models as models in which both the L-structure and the L'-structure are imposed on an object, but these structures do not interact with each other. This intuition follows from the fact that in Set a finite-product-preserving functor

$$X : L + L' \to \mathsf{Set}$$

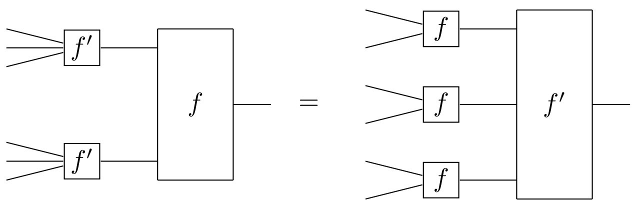
is the same are two finite-product-preserving functors

$$Y : L \to \mathsf{Set} \text{ and } Y' : L' \to \mathsf{Set}.$$

Lawvere theories can also be combined via their *tensor product*. We think of the tensor product as the theory that contains the operations of both L and L', and which imposes commutativity of the operations of L over the operations of L'. This can be made precise as follows: the Lawvere theory $L \otimes L'$ has as morphisms the disjoint union of the sets of morphisms of L and of L'. All commutative diagrams in L and L' also commute in $L \otimes L'$. In addition to these, for any $f : n \to m$ in L and $f' : n' \to m'$ in L' all the following diagrams commute

$$\begin{tikzcd}
n \times n' \arrow[r, "n \times f'"] \arrow[d, "f \times n'"'] & n \times m' \arrow[d, "f \times m'"] \\
m \times n' \arrow[r, "m \times f'"'] & m \times m'
\end{tikzcd}$$

The last condition is the commutativity of the operations of L over the operations of L'. It has a very clear string-diagrammatic description.



The tensor product has the following useful property

Theorem 3.24. *Let L and L' be Lawvere theories. The the following categories are equivalent:*

1. *The category of L-models in* $\mathsf{Mod}(L', \mathsf{Set})$*;*
2. *The category of L'-models in* $\mathsf{Mod}(L, \mathsf{Set})$*;*
3. *The category* $\mathsf{Mod}(L \otimes L', \mathsf{Set})$.

The above two methods for combining Lawvere theories are somewhat limited. To wit, these methods do not allow us to specify any "custom" equations between the operations of the two Lawvere theories. Later in this thesis we shall see that this situation can be remedied by using internal distributive laws.

3.4.3 Lawvere Theories vs Cartesian operads

In Section 3.2.4, we briefly examined the idea of Cartesian operad. As with PROs and PROPs, each Cartesian operads A gives rise to a Lawvere theory A' by setting $A'(k, 1) = A(k)$ (see Section 3.3.3). But the Cartesian structure of Lawvere Theories can be used to show that the two concepts are equivalent.

The idea behind this is easy to understand. Suppose we are given a Lawvere Theory A' and an operation $v \in A'(2, 2)$. Since A' has finite products, we can post-compose v with the projections. To obtain operations $p_1 \cdot v : 2 \to 1$ and $p_2 \cdot v : 2 \to 1$ and we have the identiry

$$v = (p_1 \cdot v) \times (p_2 \cdot v)$$

Hence, setting $\Lambda(k) = A'(k, 1)$ gives us a Cartesian operad such that the Lawvere theory associated to it - Λ' - is exactly the A' we started with.

We record this in the following Theorem:

Proposition 3.25 (Hyland)**.** *The category of Cartesian operads is equivalent to* Law*.*

The formal proof is given in [Cheng, 2011] and relies on a results we prove later in this thesis.

3.4.4 Lawvere Theories vs monads

Our treatment of Categorical Universal Algebra emphasises the use of categories with structure rather than monads for capturing the syntax of theories. And while we shall not use monads for Universal Algebraic purposes in this work, we still feel that we need to point out the connections and differences between the monadic and the Lawvere theoretic approaches.

There is a monad and a Lawvere theory for the theory of monoids: Example 2.3 shows that M-algebras are equivalent to Mon and Example 3.21 shows that the category $\mathsf{Mod}(\mathbb{M}^{\mathrm{op}}, \mathsf{Set})$ is equivalent to Mon. In general, not every monad has an associated Lawvere theory whose category of models is equivalent to the category of algebras of the monad. But in the case the monad (T, m, u) is *finitary*, that is, if it preserves *filtered colimits*, there indeed is a correspondence. We record this in a theorem.

Theorem 3.26. *There is a one-to-one correspondence between finitary monads* $(T, m.u)$ *on* Set *and Lawvere theories* L*. If* T *and* L *are matched under this correspondence then*

$$\mathsf{Mod}(L, \mathsf{Set}) \simeq \mathsf{Set}^T.$$

Example 3.27. Counter-example The powerset monad $P : \mathsf{Set} \to \mathsf{Set}$ is an example of a monad that is not finitary. The set of reals can be written as

$$\mathbb{R} = \bigcup_{A \subseteq_f \mathbb{R}} A,$$

where $\subseteq_f$ means "finite subset". The above expression is a filtered colimit of the diagram $P\mathbb{R} \hookrightarrow \mathsf{Set}$, where we view the poset $P\mathbb{R}$ as a (filtered) category and the diagram sends inclusions to inclusions. But P does not preserve this filtered colimit. Indeed, $\mathbb{R} \in P\mathbb{R}$, while

$$\mathbb{R} \in \bigcup_{A \subseteq_f \mathbb{R}} PA \implies \exists A \subseteq_f \mathbb{R} \text{ such that } \mathbb{R} \subseteq A,$$

producing the desired contradiction.

Example 3.28. Counter-example One can prove that the category of algebras of the ultrafilter monad β on Set is isomorphic to the category of compact Hausdoff spaces and continuous mappings between them. But CompHaus cannot be presented as $\mathsf{Mod}(L, \mathsf{Set})$ for any Lawvere theory L, since β does not preserve filtered colimits.

While monads provide semantics for more structures (e.g. powerset, ultrafilter monads), the algebras of each monad live over a fixed category - the domain and codomain of the functor of the monad. Lawvere theories, on the other hand, describe a more limited class of structures, but are free in the choice of the category over which these structure will live. In other words, the so-called base category does not appear in the description of the Lawvere theory.

Another important difference is in our ability to combine Lawvere theories and monads. The notion of Distributive Law is much more flexible than that of sum and tensor product of Lawvere theories. But there are no canonical distributive laws between two monads, whereas there are always at least two ways to combine Lawvere theories.

In later chapters we shall see that Lawvere theories can be seen as internal monads in certain bicategories. As such, Lawvere theories can be combined using internal distributive laws between these monads.

3.5 Concluding remarks

In this section we have explored various tools for capturing the syntax of algebraic theories of different types. All the tools we describe are clearly similar, yet all of them have to be treated as separate constructs, since, at least formally, they are different mathematical objects. We hope that the reader is as convinced as us that there is a need for a more general theory to capture all the flavours of the tools employed in Categorical Universal Algebra. Perhaps this is a sign that the Categorical Universal Algebra has reached a critical point in its development. The point where a new theory becomes broad enough to require tools for self-analysis.

One of the aims of this work is to develop such tools for operad-based Categorical Universal Algebra. We shall see that operads of all kinds are in fact special cases of a very general construction. In fact, we shall show that even the notion of 'type' or 'kind' of operad is a formal mathematical object. To make all this precise, we need to use the language of 2-categories and 2-monads which is covered in detail in the next chapters.

Chapter 4

A 2-Categorical Primer

We wish to build a framework where we can express the algebraic gadgets of the previous chapter. In Chapter 1 we saw that binary theories can be modelled by a function $t : \mathbb{N} \to 2$. To model operads, we need to replace 2 by Set. This is equivalent to categorifying the results of the first chapter. Hence instead of working with categories and functors, we wish to work with 2-categories, 2-functors, 2-natural transformations etc.

This chapter introduces the basic constructs of 2-Category Theory. We already saw higher categories in action in Chapter 2 in the Ord-enriched category Rel. Another familiar example arises when we package the important ingredients of single-dimensional category theory together. The end result is the 2-category Cat which consists of categories, functors and natural transformations.

The original definition of a strict 2-category first appeared in [Ehresmann, 1965]. Cat is presented as an example of a 2-category in [Gray, 1974]. [Bénabou, 1967] introduces bicategories, which are a more general type of 2-categories where associativity and the unit law for composition is defined up to a coherent isomorphism. Bicategories arise more often in practice than strict 2-categories. For example, every weak monoidal category is a single-object bicategory. The bicategory Prof of *profunctors* is the categorification of the category of sets and relations. It is important for describing various algebraic structures. It too is a weak 2-category.

We will begin this chapter with the definitions of strong and weak 2-categories, along with several important examples of each. We will then define 2-functors and provide examples which will be used later in the thesis. We also define 2-natural transformations and modifications. We end the chapter with a discussion of monoidal 2-categories, thus combining all the basic ingredients of 2-category theory into a single - higher-dimensional - structure. We also briefly discuss the graphical calculus for monoidal 2-categories which we shall use in proofs in later chapters.

4.1 Basic 2-categorical ingredients

4.1.1 2-categories

A *strict 2-category* can be thought of as a category in which the hom-sets are no longer simply sets, but are themselves categories. More succinctly, a strict 2-category is a CAT-enriched category. Let $\mathcal{K}$ be a strict

2-category. We call the objects of $\mathcal{K}$ *0-cells*. For any two 0-cells X and Y of $\mathcal{K}$, we call the objects of the category $\mathcal{K}(X,Y)$ *1-cells* and the morphisms of this category *2-cells*.

Let X, Y, Z be 0-cells in $\mathcal{K}$. There is a functor

$$\circ = \circ_{X,Y,Z} : \mathcal{K}(Y,Z) \times \mathcal{K}(X,Y) \to \mathcal{K}(X,Z).$$

which we call the *horizontal composition* in $\mathcal{K}$. In a strict 2-category this composition is strictly associative. Often, one wishes to avoid this strictness. When one is working with four objects X, Y, Z, W there are two passages from $\mathcal{K}(Z,W) \times \mathcal{K}(Y,Z) \times \mathcal{K}(X,Y) \to \mathcal{K}(X,W)$. Namely

$$\circ_{X,Z,W} \cdot \left(1_{\hom(Z,W)} \times \circ_{X,Y,Z}\right)$$

and

$$\circ_{X,Y,W} \cdot \left(\circ_{Y,Z,W} \times 1_{\hom(X,Y)}\right).$$

In the strict case these two functors are the same. But since these are functors, equality is only one way of comparing them. One can also ask if there is a natural isomorphism between these functors. And the latter type of comparison gives rise to the idea of a *weak 2-category* or a *bicategory*.

More precisely, a *2-category* $\mathcal{K}$ (also a *weak 2-category* or a *bicategory*) consists of the following ingredients:

1. A class of *0-cells*, denoted $X, Y, Z, \ldots$. The elements of this class are also called the *objects* of $\mathcal{K}$;
2. for each pair of objects $X, Y \in \mathcal{K}$ a category $\mathcal{K}(X,Y) = \hom(X,Y)$, whose objects are called the *1-cells* of $\mathcal{K}$ (also the *morphisms*) of $\mathcal{K}$. The morphisms of $\mathcal{K}(X,Y)$ are called the *2-cells* of $\mathcal{K}$;
3. for each object $X \in \mathcal{K}$ a functor $i_X : \mathbf{1} \to \mathcal{K}(X,X)$ called the *identity 1-cell on* X;
4. for any triple $X, Y, Z \in \mathcal{K}$ a functor

   $$\circ = \circ_{X,Y,Z} : \mathcal{K}(Y,Z) \times \mathcal{K}(X,Y) \to \mathcal{K}(X,Z).$$

   called *horizontal composition*.
5. For each $X, Y \in \mathcal{K}$ natural isomorphisms

   $$l_{X,Y} : \circ_{X,X,Y} \cdot (1_{\mathcal{K}(X,Y)} \times i_X) \to 1_{\mathcal{K}(X,Y)}$$

   and

   $$r_{X,Y} : \circ_{X,X,Y} \cdot (1_{\mathcal{K}(X,Y)} \times i_X) \to 1_{\mathcal{K}(X,Y)};$$

6. for each $X, Y, Z, W \in \mathcal{K}$ natural isomorphisms

   $$\alpha_{X,Y,Z,W} : \circ_{X,Z,W} \cdot (1_{\mathcal{K}(Z,W)} \times \circ_{X,Y,Z}) \to \circ_{X,Y,Z} \cdot (\circ_{Y,Z,W} \times 1_{\mathcal{K}(X,Y)})$$

   such that the well known pentagon and triangle diagrams commute. See [Bénabou, 1967] for the original definition and [Lack, 2007] for a modern treatment.

Example 4.1. A single object 2-category $\mathcal{K}$ is a monoidal category $K := \mathcal{K}(*,*)$. The 1-cells in $\mathcal{K}$ become the objects of K and the two cells the morphisms. Horizontal composition gives the monoidal tensor.

Example 4.2. Cat is a strict 2-category.

Example 4.3. Let $\mathcal{C}$ be any category with pullbacks. We define the bicategory $\mathsf{Span}(\mathcal{C})$ of spans in $\mathcal{C}$ by taking the object of $\mathcal{C}$ to be the 0-cells, a 1-cell from X to Y is a span, i.e., it is a triple $(E, f : E \to X, g : E \to Y)$ with $E \in \mathrm{ob}\mathcal{C}$ and f, g $\mathcal{C}$-morphisms. We usually depict spans as follows

$$\begin{tikzcd} X & & Y \\ & E \arrow[ul, "f"] \arrow[ur, "g"'] & \end{tikzcd}$$

A 2-cell from (E, f, g) to (E', f', g') is a $\mathcal{C}$-morphism $\alpha : E \to E'$ making the following two triangles commute

$$\begin{tikzcd} & E \arrow[dl, "f"'] \arrow[dr, "g"] \arrow[dd, "\alpha"] & \\ X & & Y \\ & E' \arrow[ul, "f'"] \arrow[ur, "g'"'] & \end{tikzcd} \tag{4.1}$$

We write E for the 1-cell (E, f, g) when this does not create confusion.

1-cells compose by pullback: given $E : X \to Y$ and $E' : Y \to Z$ the composite $E' \cdot E : X \to Z$ is given by the following diagram

$$\begin{tikzcd} X & & Y & & Z \\ & E \arrow[ul] \arrow[ur] & & E' \arrow[ul] \arrow[ur] & \\ & & E' \cdot E \arrow[ul] \arrow[ur] & & \end{tikzcd}$$

where the square in the middle is a pullback. The identity 1-cell on X is just $(X, 1_X, 1_X) : X \to X$. 2-cells are composed in the obvious way. We shall use $\otimes$ to denote composition of 1-cells, i.e. $E' \otimes E := E' \cdot E$.

Example 4.4. A *thin 2-category* is another name for an Ord-enriched category. Here each hom-set has a preorder on it. Thus there is at most one 2-cell between any two 1-cells, justifying the name 'thin'. For example, Rel is a thin 2-category. The two cells are given by

$$r \to s \iff r \subseteq s.$$

Ord is also a thin category; the order on hom-sets is defined pointwise: $f \leq g \iff \forall x \in X, \quad f(x) \leq g(x)$.

Example 4.5. The bicategory Prof is an important example of weak a 2-category. The objects of Prof are categories. Morphisms $X \to Y$ are given by functors

$$Y^{\mathrm{op}} \times X \to \mathsf{Set}.$$

and 2-cells are natural transformations between such morphisms. I.e.

$$\mathsf{Prof}(X, Y) = \mathsf{Cat}(Y^{\mathrm{op}} \times X, \mathsf{Set}).$$

We typically write $X \nrightarrow Y$ for a profunctor from X to Y.

Given two profunctors $\varphi : X \nrightarrow Y$ and $\psi : Y \nrightarrow Z$, the composite $\psi \cdot \varphi : X \nrightarrow Z$ is given by the coend formula[1]

$$\begin{aligned} \psi \cdot \varphi(z, x) &= \int^{y \in Y} \psi(z, y) \times \varphi(y, x) \\ &\simeq \coprod_{y \in Y} \psi(z, y) \times \varphi(y, x) / \sim \end{aligned} \tag{4.2}$$

where the relation $\sim$ is defined by saying that the pair $(f, g) \in \psi(z, y) \times \varphi(y, x)$ is equivalent to $(f', g') \in \psi(z, y') \times \varphi(y', x)$ if, and only if, there exists an $h : y \to y'$ in Y such that

$$\psi(z, h)(f) = f' \text{ and } \varphi(h, g') = g.$$

The author finds it helpful to think of a profunctor $f : X \nrightarrow Y$ as a method of assigning morphisms to pairs $(x \in X, y \in Y)$. Then the equivalence relation in 4.2 provides a way of composing such morphisms over the morphisms of Y. We shall come back to this view of profunctors in Chapter 6 when we study various monads in Prof. One can also think of profunctors are generalised relations. Every set X is a **2**-enriched category. A relation is a function

$$X^{\mathrm{op}} \times X = X \times X \to 2.$$

Hence profunctors and functors are related in the way relations and functions are.

The unit for horizontal composition is $1_X : X \nrightarrow X$ is

$$X(-, -) : X^{op} \times X \to \mathsf{Set}.$$

For $\varphi : X \nrightarrow Y$ one easily computes

$$\begin{aligned} \varphi \cdot 1_x(y, x) &= \int^{x' \in X} \varphi(y, x') \times X(x', x) \\ &\simeq \varphi(y, x) \end{aligned}$$

by the Yoneda Lemma.

Since composite profunctors are defined via a coend, the expression $\psi \cdot \varphi$ is unique only up to isomorphism. This makes Prof our first example of a non-strict 2-category.

[1] Coends are universal di-natural transformations. See Chapter IX in [Mac Lane, 1998] for a detailed treatment.

4.1.2 Pasting diagrams

We shall often use pasting diagrams to describe composites of 2-cells in a 2-category. Let $\mathcal{K}$ be a 2-category and let

$$\begin{tikzcd} X \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & Y \end{tikzcd}$$

be a 2-cell in $\mathcal{K}$. One can *whisker* α on the left and on the right to form 2-cells

$$h \circ \alpha := \begin{tikzcd} X \arrow[r, "h"] & X \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & Y \end{tikzcd} \text{ and } \alpha \circ h' := \begin{tikzcd} X \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & Y \arrow[r, "h'"] & Y \end{tikzcd}$$

We can now compose two cells such as the ones above both horizontally and vertically to form *pasting diagrams*. For example, the diagram

$$\begin{tikzcd} X \arrow[rr, "f"] \arrow[dr, "g"'] & \arrow[d, phantom, "\Downarrow\alpha"] & Y \arrow[dr, "h"] & \\ & Z \arrow[ur, "t"'] \arrow[rr, "s"'] & \arrow[u, phantom, "\Downarrow\beta"] & W \end{tikzcd} \tag{4.3}$$

stands for the vertical composition of

$$\begin{tikzcd} X \arrow[r, "g"'] \arrow[rr, bend left, "f"] & Z \arrow[r, "t"'] \arrow[u, phantom, "\Downarrow\alpha"] & Y \arrow[r, "h"'] & W \end{tikzcd} \text{ and } \begin{tikzcd} X \arrow[r, "g"] & Z \arrow[r, "t"] \arrow[rr, bend right, "s"'] & Y \arrow[r, "h"] \arrow[d, phantom, "\Downarrow\beta"] & W \\ & & {} & \end{tikzcd} \tag{4.4}$$

Note that in writing

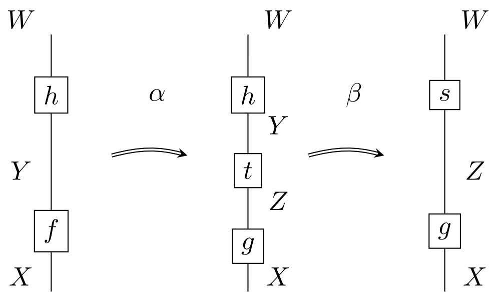
$$X \xrightarrow{g} Z \xrightarrow{t} Y \xrightarrow{h} W$$

we have made an explicit choice for the bracketing of this composite. Thus to compose the 2-cells of 4.4 we need to introduce an associativity isomorphism $h \circ (t \circ g) \implies (h \circ t) \circ g$. But, as is often done in practice, we shall ignore the structure of the underlying 2-category $\mathcal{K}$ in our pasting diagrams.

Pasting diagrams can be written in many forms. For example, the diagram of 4.3 can be written as

$$h \circ f \xrightarrow{\alpha \circ h} h \circ t \circ g \xrightarrow{g \circ s} g \circ s$$

equivalently, this data cane written graphically as



Note that the graphical depiction also captures the 0-cells as strings feeding in and out of the 1-cells. The latter representation of pasting diagrams allows us to think of 2-cells as rewrites of diagrams. We shall come back to this view often throughout this work.

4.1.3 2-functors

Let $\mathcal{K}$ and $\mathcal{M}$ be 2-categories. A *2-functor*

$$F : \mathcal{K} \to \mathcal{M}$$

consists of the following data

1. a mapping F from the 0-cells of $\mathcal{K}$ to the 0-cells of $\mathcal{M}$;
2. for any two objects $X, Y \in \mathcal{K}$ a functor
$$F_{X,Y} : \mathcal{K}(X,Y) \to \mathcal{M}(FX, FY);$$
3. for each 0-cell $X \in \mathcal{K}$ an invertible 2-cell in $\mathcal{M}$
$$F(id)_X : 1_{FX} \to F_{X,X}(1_X);$$
4. for any pair of composable 1-cells $X \xrightarrow{f} Y \xrightarrow{g} Z \in \mathcal{K}$ an invertible 2-cell in $\mathcal{M}$
$$F(g,f) : F_{Y,Z}(g) \cdot F_{X,Y}(f) \to F_{X,Z}(g \cdot f)$$
such that $F(f,g)$ is natural in both f and g;
5. such that for any three 1-cells $f : X \to Y$, $g : Y \to Z$ and $h : Z \to W$ in $\mathcal{K}$
$$\begin{tikzcd}
F1_Y \cdot Ff \arrow[r, "F(1_Y)^{-1}"] \arrow[d, "F(1_Y,f)"'] & 1_{FY} \cdot Ff \arrow[d, "l_{FX,FY}"] \\
F(1_Y \cdot f) \arrow[r, "F(l_{X,Y})"'] & Ff
\end{tikzcd}
\qquad
\begin{tikzcd}
Fg \cdot F1_Y \arrow[r, "Fg \cdot F(1_Y)^{-1}"] \arrow[d, "F(g,1_Y)"'] & Fg \cdot 1_{FY} \arrow[d, "r_{FY,FZ}"] \\
F(g \cdot 1_Y) \arrow[r, "F(r_{Y,Z})"'] & Fg
\end{tikzcd}$$
and
$$\begin{tikzcd}
F(h) \cdot \big(F(g) \cdot F(f)\big) \arrow[r, "\alpha"] \arrow[d, "1 \cdot F(g,f)"'] & \big(F(h) \cdot F(g)\big) \cdot F(f) \arrow[d, "F(h,g) \cdot 1"] \\
F(h) \cdot F(g \cdot f) \arrow[d, "F(h, g \cdot f)"'] & F(h \cdot g) \cdot F(f) \arrow[d, "F(h \cdot g), f)"] \\
F\big(h \cdot (g \cdot f)\big) \arrow[r, "F\alpha"'] & F\big((h \cdot g) \cdot f\big)
\end{tikzcd}$$

One can relax the definition of a 2-functor by removing the requirement that the morphisms $F(g,f)$ and $F(id)_X$ are isomorphisms. In this case we call F a *lax functor*. If the directions of these morphisms are reversed, F becomes an *op-lax functor*.

Lets examine what it means for $F(g,f)$ to be natural in both f and g. The two passages

$$\begin{tikzcd}
\mathcal{K}(Y,Z) \times \mathcal{K}(X,Y) \arrow[r, "\circ"] \arrow[d, "F\times F"'] & \mathcal{K}(X,Z) \arrow[d, "F"] \\
\mathcal{M}(FY,FZ) \times \mathcal{M}(FX,FY) \arrow[r, "\circ"] & \mathcal{M}(FX,FZ)
\end{tikzcd}$$

are both functors. The bottom-left passage sends a pair (g,f) to $F_{Y,Z}(g)\cdot F_{X,Y}(f)$ and the top-right passage sends it to $F_{X,Z}(g\cdot f)$. So, naturality amounts to saying that for any pair of 2-cells $\alpha : f \to f'$ and $\beta : g \to g'$ in $\mathcal{K}$, the following diagram commutes

$$\begin{tikzcd}
F_{Y,Z}(g)\cdot F_{X,Y}(f) \arrow[r, "F(g,f)"] \arrow[d, "F_{Y,Z}(\beta)\circ F_{X,Y}(\alpha)"'] & F_{X,Z}(g\cdot f) \arrow[d, "F_{X,Z}(\beta\circ\alpha)"] \\
F_{Y,Z}(g')\cdot F_{X,Y}(f') \arrow[r, "F(g',f')"'] & F_{X,Z}(g'\cdot f')
\end{tikzcd}$$

Thus naturality in condition (4) of the definition ensures that composition and functoriality take the 2-dimensional structures of $\mathcal{K}$ and $\mathcal{M}$ into account.

Example 4.6. Our first example is the categorification of the underlying functor of the monoid monad. We define a 2-functor $M : \mathsf{Cat} \to \mathsf{Cat}$ as follows.

1. On 0-cells, i.e. on categories, $M(\mathcal{C})$ is the free monoidal category on $\mathcal{C}$. That is,

$$\mathrm{ob}M(\mathcal{C}) = \{A_1 \otimes \ldots \otimes A_n \mid A_i \in \mathrm{ob}\mathcal{C}\}.$$

   When $m = n$,

$$M(\mathcal{C})(A_1 \otimes \ldots \otimes A_n, B_1 \otimes \ldots \otimes B_m) = \{f_i \otimes \ldots \otimes f_n \mid f_i \in \mathcal{C}(A_i, B_i)\},$$

   and is $\emptyset$ otherwise.

2. For any two categories $\mathcal{C}$ and $\mathcal{D}$ we define

$$M_{\mathcal{C},\mathcal{D}} : \mathsf{Cat}(\mathcal{C},\mathcal{D}) \to \mathsf{Cat}(M(\mathcal{C}), M(\mathcal{D})),$$

   by, for any $F : \mathcal{C} \to \mathcal{D}$, setting

$$M(F)(A_1 \otimes \ldots \otimes A_n) = F(A_1) \otimes \ldots \otimes F(A_n),$$

   and similarly on morphisms. For a natural transformation $\alpha : F \to G$ we (are forced to) set

$$(M\alpha)_{A_1\otimes\ldots\otimes A_n} = \alpha_{A_1} \otimes \ldots \otimes \alpha_{A_n}.$$

3. For any category $\mathcal{C}$ the 2-cell

$$\begin{aligned} M(1_{\mathcal{C}})(X_1 \otimes \ldots \otimes X_n) &= 1_{\mathcal{C}}(X_1) \otimes \ldots \otimes 1_{\mathcal{C}}(X_n) \\ &= X_1 \otimes \ldots \otimes X_n \\ &= 1_{M(\mathcal{C})} X_1 \otimes \ldots \otimes X_n. \end{aligned}$$

   Thus the isomorphism in condition (3) is actually an equality.

4. Let $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$ be functors.

$$\begin{aligned} M_{\mathcal{D},\mathcal{E}}(G) \cdot M_{\mathcal{C},\mathcal{D}}(F)(X_1 \otimes \ldots \otimes X_n) &= M_{\mathcal{D},\mathcal{E}}(G)(FX_1 \otimes \ldots \otimes FX_n) \\ &= GFX_1 \otimes \ldots \otimes GFX_n \\ &= M_{\mathcal{C},\mathcal{E}}(G \cdot F)(X_1 \otimes \ldots \otimes X_n) \end{aligned}$$

   and similarly on morphisms. So, again, the natural isomorphism of condition (4) is a strict equality.

Functors such as M in which the isomorphisms of conditions (3) and condition (4) are equalities, are called *strict 2-functors*.

Example 4.7. Similarly, the *symmetric monoidal category functor* $S : \mathsf{Cat} \to \mathsf{Cat}$ sends every category $\mathcal{C}$ to the free symmetric monoidal category on $\mathcal{C}$.

Example 4.8. Another important example is the *finite product functor* $F : \mathsf{Cat} \to \mathsf{Cat}$. F sends a category $\mathcal{C}$ to the free finite product category on $\mathcal{C}$.

Example 4.9. There is a 2-functor

$$(-)_* : \mathsf{Cat} \to \mathsf{Prof}$$

which is identity on objects and send every functor $F : \mathcal{C} \to \mathcal{D}$ to the profunctor

$$F_* := \mathcal{D}(-, F(-)) : \mathcal{D}^{\mathrm{op}} \times \mathcal{C} \to \mathsf{Set}.$$

There is a 2-functor

$$(-)^* : \mathsf{Cat}^{\mathrm{op}} \to \mathsf{Prof}$$

that is identity on objects and sends every functor $F : \mathcal{C} \to \mathcal{D}$ in Cat to the profunctor

$$F^* := \mathcal{D}(F(-), -) : \mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathsf{Set}.$$

Example 4.10. The presheaf functor

$$P : \mathsf{Cat} \to \mathsf{CAT}$$

sends every category $\mathcal{C}$ to its colimit cocompletion $P\mathcal{C} = \hat{\mathcal{C}} := \mathsf{Set}^{\mathcal{C}^{\mathrm{op}}}$. Let $F : \mathcal{C} \to \mathcal{D}$ be a functor and $\mathcal{G} : \mathcal{C}^{\mathrm{op}} \to \mathsf{Set}$ a presheaf. Define the preshaef $PF(\mathcal{G}) : \mathcal{D}^{\mathrm{op}} \to \mathsf{Set}$ as the left Kan extension of $\mathcal{G}$ along

F^{op}: $PF(\mathcal{G}) := \mathsf{Lan}_{F^{\mathrm{op}}}(\mathcal{G})$.

$$\begin{tikzcd}
\mathcal{C}^{\mathrm{op}} \arrow[rr, "F^{\mathrm{op}}"] \arrow[dr, "\mathcal{G}"'] & \arrow[d, phantom, "\overset{k}{\Rightarrow}"] & \mathcal{D}^{\mathrm{op}} \arrow[dl, dashed, "PF(\mathcal{G})"] \\
& \mathsf{Set} &
\end{tikzcd}$$

We express the action of $PF(\mathcal{G})$ on an object X of $\mathcal{D}$ via the standard coend formula

$$PF(\mathcal{G})(X) = \int^{Y\in\mathcal{C}} \mathcal{G}(Y) \times \mathcal{D}(X, FY).$$

Let $\alpha : F \to H$ be a natural transformation. We define $P\alpha : PF \to PH$ at every presheaf $\mathcal{G} \in P\mathcal{C}$ by exploiting the universal property of left Kan extensions. Consider the following composite

$$\mathcal{G} \xrightarrow{k} PH(\mathcal{G}) \cdot H^{\mathrm{op}} \xrightarrow{1\cdot\alpha^{\mathrm{op}}} PH(\mathcal{G}) \cdot F^{\mathrm{op}};$$

it allows us to derive, via the universal property, a unique natural transformation

$$P\alpha_{\mathcal{G}} : PF(\mathcal{G}) \to PH(\mathcal{G}).$$

Having defined the action of P on 0-, 1- and 2-cells of Cat we proceed by verifying if functoriality. Fix categories $\mathcal{C}$ and $\mathcal{D}$. Then

$$P_{\mathcal{C},\mathcal{D}} : \mathsf{Cat}(\mathcal{C}, \mathcal{D}) \to \mathsf{CAT}(\mathcal{C}, \mathcal{D})$$

defines a functor. Indeed, given

$$1_F : F \to F$$

for any $\mathcal{G} : \mathcal{C}^{\mathrm{op}} \to \mathsf{Set}$ we have

$$\begin{tikzcd}
\mathcal{G} \arrow[r, "k"] \arrow[dr, "k"'] & PF(\mathcal{G}) \cdot F^{\mathrm{op}} \arrow[r, "1"] & PF(\mathcal{G}) \cdot F^{\mathrm{op}} \\
& PF(\mathcal{G}) \cdot F^{\mathrm{op}} \arrow[ur, "1_{PF(\mathcal{G})}\cdot F^{\mathrm{op}}"'] &
\end{tikzcd}$$

showing

$$(P1_F)_{\mathcal{G}} = (1_{PF})_{\mathcal{G}}$$

by the unicity of the above factorisation.

Similarly, given two natural transformations

$$F \xrightarrow{\alpha} H \xrightarrow{\beta} G\ ,$$

we have

$$P(\alpha)_{\mathcal{G}} \cdot P(\beta)_{\mathcal{G}} = P(\alpha \cdot \beta)_{\mathcal{G}}$$

We finish by exhibiting the invertible 2-cells

$$P(id_{\mathcal{C}}) : 1_{P(\mathcal{C})} \overset{\simeq}{\Rightarrow} P(1_{\mathcal{C}})$$

and

$$P(H,F) : P(H)\cdot P(F) \stackrel{\simeq}{\Rightarrow} P(H\cdot F),$$

for $F : \mathcal{C} \to \mathcal{D}$, $H : \mathcal{D} \to \mathcal{E}$.

The first isomorphism follows from the Yoneda Lemma: for every $\mathcal{G}$ and every X

$$\mathcal{G}(X) = \int^{Y\in\mathcal{C}^{\mathrm{op}}} \mathcal{G}(Y) \times \mathcal{C}(X,Y).$$

The second one follows is given by the following identities

$$\begin{aligned}
P(H)\big(P(F)(\mathcal{G})\big)(X) &= \int^{Z\in\mathcal{D}} P(F)(\mathcal{G})(Z) \times \mathcal{E}(X, H(Z)) \\
&\simeq \int^{Z\in\mathcal{D}} \Big(\int^{Y\in\mathcal{C}} \mathcal{G}(Y) \times \mathcal{D}(Z, F(Y))\Big) \times \mathcal{E}(X, H(Z)) \\
&\simeq \int^{Y\in\mathcal{C}} \mathcal{G}(Y) \times \int^{Z\in\mathcal{D}} \mathcal{D}(Z, F(Y)) \times \mathcal{E}(X, H(Z)) \\
&\simeq \int^{Y\in\mathcal{C}} \mathcal{G}(Y) \times \mathcal{E}(X, H(F(Y))) \\
&= P(H\cdot F)(\mathcal{G})(X).
\end{aligned}$$

4.1.4 2-natural transformations

Let F and G be two 2-functors $\mathcal{K} \to \mathcal{M}$. A *2-natural transformation* (also *pseudo-natural transformation*) consists of the following data

1. for each 0-cell X of $\mathcal{K}$ a 1-cell
$$\alpha_X : FX \to GX$$
in $\mathcal{M}$.

2. for each 1-cell $f : X \to Y$ in $\mathcal{K}$ an invertible 2-cell
$$\begin{tikzcd}
FX \arrow[r, "Ff"] \arrow[d, "\alpha_X"'] \arrow[dr, phantom, "\swarrow\alpha_f"] & GY \arrow[d, "\alpha_Y"] \\
GX \arrow[r, "Gf"'] & GY
\end{tikzcd}$$

such that the following coherence conditions are satisfied:

1. for any $f : X \to Y$ and $g : Y \to Z$ in $\mathcal{K}$

$$
\begin{tikzcd}
& FY \arrow[dr, "Fg"] \arrow[dd, "\alpha_Y"] & \\
FX \arrow[ur, "Ff"] \arrow[dd, "\alpha_X"'] \arrow[dr, phantom, "\Swarrow\alpha_f"] & & FZ \arrow[dd, "\alpha_Z"] \arrow[dl, phantom, "\Swarrow\alpha_g"] \\
& GY \arrow[dr, "Gg"] \arrow[d, phantom, "\Downarrow"] & \\
GX \arrow[ur, "Gf"] \arrow[rr, "G(g\cdot f)"'] & {} & GZ
\end{tikzcd}
=
\begin{tikzcd}
& FY \arrow[dr, "Fg"] \arrow[d, phantom, "\Downarrow"] & \\
FX \arrow[ur, "Ff"] \arrow[rr, "F(g\cdot f)"] \arrow[dd, "\alpha_X"'] & {} \arrow[d, phantom, "\Downarrow\alpha_{g\cdot f}"] & FZ \arrow[dd, "\alpha_Z"] \\
& {} & \\
GX \arrow[rr, "G(g\cdot f)"'] & & GZ
\end{tikzcd}
\tag{4.5}
$$

2. for every 0-cell X of $\mathcal{K}$

$$
\begin{tikzcd}
FX \arrow[r, "\alpha_X"] \arrow[d, "F1_X"] \arrow[d, bend right=60, "1_{FX}"'] \arrow[d, phantom, bend right=30, "\simeq"] \arrow[dr, phantom, "\Downarrow\alpha_1"] & GX \arrow[d, "G1_X"'] \arrow[d, bend left=60, "1_{GX}"] \arrow[d, phantom, bend left=30, "\simeq"] \\
FX \arrow[r, "\alpha_X"'] & GX
\end{tikzcd}
=
\begin{tikzcd}
FX \arrow[r, "\alpha_X"] \arrow[d, "1_{FX}"'] \arrow[dr, "\alpha_X"] & GX \arrow[d, "1_{GX}"] \\
FX \arrow[r, "\alpha_X"'] & GX
\end{tikzcd}
\tag{4.6}
$$

3. for any two cell

$$
\begin{tikzcd}
X \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\gamma"] & Y
\end{tikzcd}
$$

in $\mathcal{K}$

$$
\begin{tikzcd}
FX \arrow[r, bend left=70, "Ff"] \arrow[r, phantom, bend left=35, "\Downarrow F\gamma"] \arrow[r, "Fg"'] \arrow[d, "\alpha_X"'] \arrow[dr, phantom, "\Downarrow"] & FY \arrow[d, "\alpha_Y"] \\
GX \arrow[r, "Gg"'] & GY
\end{tikzcd}
=
\begin{tikzcd}
FX \arrow[r, "Ff"] \arrow[d, "\alpha_X"'] \arrow[dr, phantom, "\Downarrow"] & FY \arrow[d, "\alpha_Y"] \\
GX \arrow[r, "Gf"] \arrow[r, bend right=70, "Gg"'] \arrow[r, phantom, bend right=35, "\Downarrow G\gamma"] & GY
\end{tikzcd}
\tag{4.7}
$$

Example 4.11. There is a strict 2-natural transformation, j, from the symmetric monoidal category functor, S, to the finite-product category functor, F. At each category X,

$$j_X : SX \hookrightarrow FX$$

is the functor that sends formal tensors to formal products:

$$j_X(x_1 \otimes \ldots \otimes x_n) = x_1 \times \ldots \times x_n$$

Example 4.12. The Yoneda embedding, $y : 1_{\mathsf{Cat}} \to P$ is a 2-natural transformation. At each category

$$y_X : X \to PX, \text{ sends } x \in X \text{ to } \hom(-, x).$$

The pseudo-naturality

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "y_X"'] \arrow[dr, phantom, "\Downarrow"] & Y \arrow[d, "y_Y"] \\
PX \arrow[r, "Pf"'] & PY
\end{tikzcd}$$

can be derived from the Density formula for presheaves:

$$\begin{aligned}
(Pf \cdot y_X(x))(y) &= \int^{x' \in X} y_X(x, x') \times \hom_Y(y, f(x')) \\
&\simeq \int^{x' \in X} \hom_Y(y, f(x')) \times \hom_X(x', x) \\
&\simeq \hom_Y(y, f(x)) \\
&= \big(y_Y \cdot f(x)\big)(y)
\end{aligned}$$

4.1.5 Modifications

Let $\alpha, \beta : F \to G$ be two 2-natural transformations between 2-functors F and G. A *modification* $\gamma : \alpha \to \beta$ consists of a collection of 2-cells

$$\{\gamma_X : \alpha_X \to \beta_X \mid X \in \mathrm{ob}\mathcal{K}\}$$

such that for every 1-cell $f : X \to Y$ in $\mathcal{K}$

$$\begin{tikzcd}
FX \arrow[r, "\alpha_X"] \arrow[d, "Ff"'] \arrow[dr, phantom, "\Downarrow"] & GX \arrow[d, "Gf"] \\
FY \arrow[r, "\alpha_Y"] \arrow[r, bend right=60, "\beta_Y"'] \arrow[r, phantom, shift right=3, "\Downarrow\gamma_Y"] & GY
\end{tikzcd}
=
\begin{tikzcd}
FX \arrow[r, "\beta_X"'] \arrow[r, bend left=60, "\alpha_X"] \arrow[r, phantom, shift left=3, "\Downarrow\gamma_X"] \arrow[d, "Ff"'] \arrow[dr, phantom, "\Downarrow"] & GX \arrow[d, "Gf"] \\
FY \arrow[r, "\beta_Y"'] & GY
\end{tikzcd}$$

Example 4.13. There are two strict 2-natural transformations

$$u, \kappa : 1_{\mathsf{Cat}} \to F$$

given by

$$u_X : X \to FX \text{ which sends } x \mapsto x \text{ and } \kappa_X : X \to FX \text{ which sends } x \mapsto x \times x.$$

Note that for every category X, the components u_X and κ_X are functors. We can define a modification $\Delta : u \to \kappa$. For every category X, the natural transformation

$$\Delta_X : u_X \to \kappa_X$$

is defined taking $(\Delta_X)_x : x \to x \times x$ to be the diagonal in FX. The naturality of Δ_X in x follows from the universal property of the product $x \times x$ in FX. It remains to show that Δ is a modification. Since our 2-natural transformations are strict, the condition reduces to showing that for every functor $f : X \to Y$

$$\begin{tikzcd} X \arrow[r, "f"] & Y \arrow[r, bend left, "u_Y"] \arrow[r, bend right, "\kappa_Y"'] \arrow[r, phantom, "\Downarrow \Delta_Y"] & FY \end{tikzcd} = \begin{tikzcd} X \arrow[r, bend left, "u_X"] \arrow[r, bend right, "\kappa_X"'] \arrow[r, phantom, "\Downarrow \Delta_X"] & FX \arrow[r, "Ff"] & FY \end{tikzcd}$$

and both sides of the equation equal $(\Delta_Y)_{f(x)}$.

4.2 Relationships between 2-categories

In this section we define 2-adjunctions and 2-equivalences. These concepts capture important relationships between two categories and shall be used to this end in upcoming chapters.

4.2.1 Biequivalence

We say that two 2-categories $\mathcal{K}$ and $\mathcal{K}'$ are *biequivalent* or *2-equivalent* provided that there exist 2-functors

$$F : \mathcal{K} \to \mathcal{K}' \text{ and } G : \mathcal{K}' \to \mathcal{K}$$

and 2-natural transformations $i : G \cdot F \to 1_{\mathcal{K}}$ and $j : F \cdot G \to 1_{\mathcal{K}'}$ which are themselves equivalences *in* the 2-categories End($\mathcal{K}$) and End($\mathcal{K}'$). This means, for example for i, that there exists another 2-natural transformation $i' : 1_{\mathcal{K}} \to G \cdot F$ and an invertible modifications

$$\xi : i \cdot i' \to 1_{1_{\mathcal{K}}} \text{ and } \xi' : i' \cdot i \to 1_{G \cdot F}.$$

See [Schreiber, 2011] for more details. We shall give examples of biequivalent 2-categories later in the thesis.

4.2.2 2-Adjunctions

There are several types of adjunctions between 2-categories. We begin with *lax 2-adjunctions*. Let $\mathcal{K}$ and $\mathcal{M}$ be 2-categories, let $F : \mathcal{K} \to \mathcal{M}$ and $G : \mathcal{M} \to \mathcal{K}$ be 2-functors. Wa say that F is *left lax 2-adjoint* to G provided that for all 0-cells $X \in \mathcal{K}$ and $Y \in \mathcal{M}$ there exist adjunctions

$$\begin{tikzcd} \mathcal{M}(FX, Y) \arrow[r, shift left, "f_{X,Y}"] \arrow[r, phantom, "\perp"] & \mathcal{K}(X, GY) \arrow[l, shift left, "g_{X,Y}"] \end{tikzcd} \tag{4.8}$$

2-natural is X and Y. The last condition means that the functors $f_{X,Y}$ and $g_{X,Y}$ are components of 2-natural transformations $f_{-,Y}$, $f_{X,-}$, $g_{-,Y}$, $g_{X,-}$.

We can strengthen condition 4.8 and ask for that for the functors $f_{X,Y}$ and $g_{X,Y}$ to form an equivalence. The resulting relationship between F and G is called a *pseudo-adjunciton*. Of course, $f_{X,Y}$ and $g_{X,Y}$ still have to be natural in X and Y.

An example of a important pseudo-adjunction is given in Example 5.5. Lax 2-adjunctions play a central role in Theorem **??** and in its applications in Chapter **??**

4.3 Monoidal 2-categories

2-categories, 2-functors, 2-natural transformations and modifications form a so-called *tricategory* (See Section 6.3 in [Gurski, 2007]). A *monoidal bicategory* is a single object tricategory. That is, it is a 2-category $\mathcal{K}$ equipped with a 2-functor

$$\otimes : \mathcal{K} \times \mathcal{K} \to \mathcal{K}$$

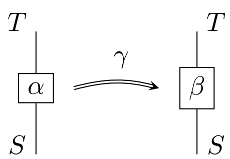
and a 0-cell $I \in \mathcal{K}$ acting as a unit for $\otimes$. The complete definition is complex and can be found in [Gordon et al., 1995]. We use monoidal bicategories mainly to organise data in our constructions and for these purposes we do not need the full details of the formal definition.

We shall rely on the fact that given any 2-categories $\mathcal{K}$ and $\mathcal{M}$, the bicategory $\mathsf{BiCat}(\mathcal{K}, \mathcal{M})$ is (triequivalent to) a semi-strict monoidal 2-category [Gurski, 2007]. Recall from [Street, 2003] that such bicategories admit a graphical calculus. In that paper, Street depics 2-cells in a monoidal bicategory $\mathcal{K}$ as surfaces in 3-dimensional Euclidian space. We cut his surfaces into 2-dimensional slices and depict 2-cells as rewrites between these slices. We give details below.

Let $\mathcal{K}$ be a monoidal 2-category $\mathcal{K}$. Since $\mathcal{K}$ is in particular a 2-category, we can use pasting diagrams to represent data in $\mathcal{K}$. In particular, we still depict any 2-cell as

$$\begin{tikzcd} S \arrow[r, bend left, "\alpha"] \arrow[r, bend right, "\beta"'] \arrow[r, phantom, "\Downarrow\gamma"] & T \end{tikzcd}$$

or, equivalently, as

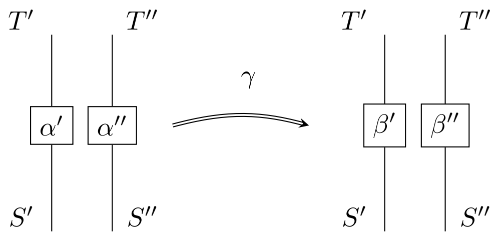


in our graphical calculus.

Tensor products are depicted by placing diagrams for components of the tensor side-by-side. For example, the pasting diagram

$$\begin{tikzcd} S' \otimes S'' \arrow[r, bend left, "\alpha' \otimes \alpha''"] \arrow[r, bend right, "\beta' \otimes \beta''"'] \arrow[r, phantom, "\Downarrow\gamma"] & T' \otimes T'' \end{tikzcd}$$

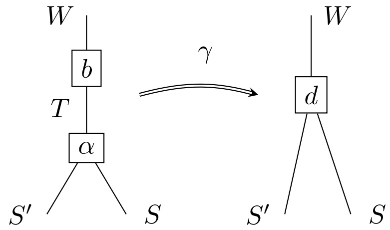
translates to



Horizontal composites of 1-cells are represented by stacking to boxes corresponding to these cells on top of each other. For example, the pasting diagram

$$\begin{tikzcd} S' \otimes S \arrow[r, "a"] \arrow[rr, bend right, "d"'] & T \arrow[r, "b"] \arrow[d, phantom, "\Downarrow\gamma"] & W \\ & {} & \end{tikzcd}$$

is represented by

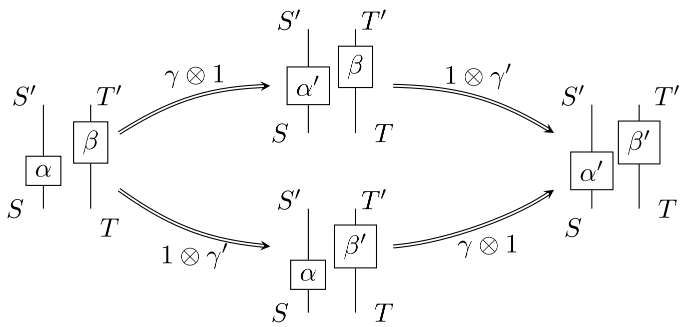


Now that we are familiar with the basic constructs, we establish some simple rules for manipulating this graphical representation.

Proposition 4.14. *Suppose we are give two 2-cells in* $\mathsf{BiCat}(\mathcal{K}, \mathcal{M})$*:*

$$\begin{tikzcd} S \arrow[r, bend left, "\alpha"] \arrow[r, bend right, "\beta"'] \arrow[r, phantom, "\Downarrow\gamma"] & T \end{tikzcd} \qquad \begin{tikzcd} S' \arrow[r, bend left, "\alpha'"] \arrow[r, bend right, "\beta'"'] \arrow[r, phantom, "\Downarrow\gamma'"] & T' \end{tikzcd}$$

then



Proof. In any monoidal bicategory both passages are equal to

$$\gamma \otimes \gamma' : \alpha \otimes \beta \to \alpha' \otimes \beta'$$

□

Modifications can be viewed as processes that *transform* or *rewrite* one diagram into another. The above proposition says that when it comes to independent rewrites, the order of executing does not make any difference.

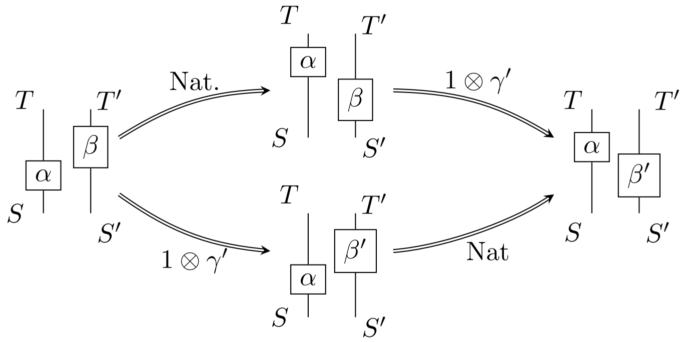
Proposition 4.15. *Suppose we are given a 1-cell*

$$S \xrightarrow{\alpha} T$$

and a 2-cell

$$\begin{tikzcd} S' \arrow[r, bend left, "\beta"] \arrow[r, bend right, "\beta'"'] \arrow[r, phantom, "\Downarrow\gamma'"] & T' \end{tikzcd}$$

$\mathcal{K}$. Then



Proof. This follows directly from the naturality condition on the pseudo-naturality of α. $\square$

The above Proposition says that first reordering the boxes and then applying the rewrite is the same process as first rewriting and then reordering the boxes.

The above two propositions will prove instrumental in verifying complex coherence conditions in the next chapters.

Chapter 5

2-monad theory

In this Chapter we continue exploring the 2-categorical machinery we need to study the algebraic gadgets of Chapter 3. Recall from Chapter 2 that we used monads on Set and distributive laws between them to formulate a framework where we defined binary theories of various types. To do this for more interesting theories, we need to categorify that setting. This requires to shift from Set-monads to 2-monads on Cat and 2-distributive laws between them.

We begin the chapter with a definition of a 2-monad. A 2-monad consist of a 2-functor T along with 2-natural transformations $m : T^2 \to T$ and $u : 1 \to T$ such that the monad axioms hold up to coherent isomorphisms. In our treatment, we attempt to create an intuitive feeling for the meaning of these coherence conditions. To do this, we formulate coherence using the graphical language developed in the previous chapter. We also analyse various strict and weak 2-monads on Cat. In particular we categorify both the monoid monad M and the powerset monad P of Chapter 2.

M can be categorifyed in many ways. The easiest is via the free monoidal category construction. In this construction the we do not add any new morphisms to the image MX. A more enticing option is to add symmetry morphisms to the mix. This gives rise to the symmetric monoidal category monad. Finally, we can freely add finite product structure creating the finite-product category monad. We shall see in the next chapter that these three monads will help us to define plain, symmetric and Cartesian operads.

In Chapter 2 we saw that binary theories are internal monads in Kleisli categories. For 2-monads we need the notion of a Kleisli *bi*category. We present the definition given of [Cheng et al., 2004] and list some properties of this construction.

Having understood 2-monads, we move to 2-distributive laws. Again, our focus is on the examples. We show that the monoidal, symmetric monoidal and Cartesian category monads distribute over the presheaf monad and explicitly derive the formulas for these distributive laws. Using a 2-categorical version of Beck's Theorem, we obtain extensions of all our examples to Prof. We conclude the chapter with an analysis of the actions of these extensions on a given profunctor.

2-monads are a fairly modern development. Kelly's work on enriched category theory [Kelly, 1982] can be specialised to give Cat-enriched monad theory. This gives the theory of strict 2-monads and strict algebras. In [Blackwell et al., 1989], the authors develop the theory of weak 2-monads which we use here. The Kleisli bicategory construction we use was developed in [Cheng et al., 2004]. 2-distributive laws were defined in [Marmolejo, 1999]. Beck's Theorem for 2-monads and 2-distributive laws is proved in [Cheng et al., 2004].

5.1 2-monads

Let $\mathcal{K}$ be a 2-category. A *2-monad* on $\mathcal{K}$ is a 6-tuple $(S, m, u, \tau, \lambda, \rho)$ where $S : \mathcal{K} \to \mathcal{K}$ is a 2-functor, $m : S^2 \to S$ and $u : 1_{\mathcal{K}} \to S$ are 2-natural transformations and τ, λ and ρ are invertible modifications

$$\begin{tikzcd} S^3 \arrow[r, "mS"] \arrow[d, "Sm"'] \arrow[dr, phantom, "\Downarrow\tau"] & S^2 \arrow[d, "m"] \\ S^2 \arrow[r, "m"] & S \end{tikzcd}$$

and

$$\begin{tikzcd} S \arrow[r, "Su"] \arrow[dr, "1_S"'] & S^2 \arrow[d, "m"] \arrow[dl, phantom, "\overset{\rho}{\Rightarrow}"] \arrow[dr, phantom, "\overset{\lambda}{\Leftarrow}"] & S \arrow[l, "uS"'] \arrow[dl, "1_S"] \\ & S & \end{tikzcd}$$

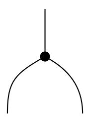
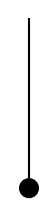
subjects to the following coherence axioms

$$\begin{tikzcd}[column sep=small] S^4 \arrow[rr, "S^2m"] \arrow[dr, "SmS"] \arrow[dd, "mS^2"'] & \arrow[d, phantom, "\Downarrow S\tau"] & S^3 \arrow[dr, "Sm"] & \\ \arrow[d, phantom, "\Downarrow\tau S"] & S^3 \arrow[rr, "Sm"] \arrow[dd, "mS"] & \arrow[d, phantom, "\Downarrow\tau"] & S^2 \arrow[dd, "m"] \\ S^3 \arrow[dr, "mS"'] & & & \\ & S^2 \arrow[rr, "m"'] & & S \end{tikzcd} = \begin{tikzcd}[column sep=small] S^4 \arrow[rr, "S^2m"] \arrow[dd, "mS^2"'] & \arrow[d, phantom, "\Downarrow m_m"] & S^3 \arrow[dr, "Sm"] \arrow[dd, "mS"'] & \\ & & \arrow[r, phantom, "\Downarrow\tau"] & S^2 \arrow[dd, "m"] \\ S^3 \arrow[rr, "Sm"] \arrow[dr, "mS"'] & \arrow[d, phantom, "\Downarrow\tau"] & S^2 \arrow[dr, "m"] & \\ & S^2 \arrow[rr, "m"] & & S \end{tikzcd}$$

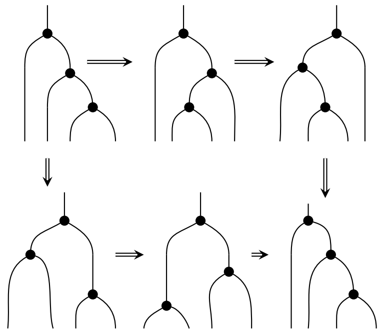
$$\begin{tikzcd} S^2 \arrow[dr, "SuS"] \arrow[rr, bend left, "1_{S^2}"] \arrow[d, bend right=60, "1_{S^2}"'] & \arrow[d, phantom, "\Downarrow S\rho"] & \\ \arrow[r, phantom, "\Downarrow\lambda S"] & S^3 \arrow[r, "mS"] \arrow[d, "Sm"] \arrow[dr, phantom, "\Downarrow\tau"] & S^2 \arrow[d, "m"] \\ & S^2 \arrow[r, "m"'] & S \end{tikzcd} \quad = \quad \begin{tikzcd} S^2 \arrow[r, "m"] & S \end{tikzcd}$$

We say that a 2-monad is *strict* when the modifications τ, λ and ρ are equalities.

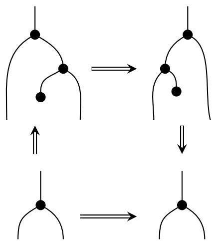
Since all the data for a 2-monad $S : \mathcal{K} \to \mathcal{K}$ lives in the monoidal bicategory $\mathrm{End}(\mathcal{K})$ we can represent this data using graphical calculus 4.3. We write $m : S^2 \to S$ and $u : 1_{\mathcal{K}} \to S$ as

$m =$  $u =$ 

the graphical representation gives rise to an intuitive understanding the coherence conditions:



and



We give several examples of 2-monads that we shall encounter later in the thesis.

Example 5.1. Let M be the 2-functor of Example 4.6. M is part of a 2-monad $(M, m, u, =, =, =)$; we shall use M to highlight some of the hidden technical issues in the definition of a 2-monad.

The multiplication is given by the 1-cell $m_{\mathcal{C}} : MM\mathcal{C} \to M\mathcal{C}$ in Cat. That is, for every $\mathcal{C}$, $m_{\mathcal{C}}$ is a functor. Let

$$(x_{11} \otimes \ldots \otimes x_{1n_1}) \otimes \ldots \otimes (x_{k1} \otimes \ldots \otimes x_{kn_k})$$

be an object of $MM\mathcal{C}$. The functor $m_{\mathcal{C}}$ sends it to

$$x_{11} \otimes \ldots \otimes x_{1n_1} \otimes \ldots \otimes x_{k1} \otimes \ldots \otimes x_{kn_k}$$

in $M\mathcal{C}$. That is, $m_{\mathcal{C}}$ just concatenates the inner brackets. Its action on morphisms of $MM\mathcal{C}$ is exactly the same. Functoriality of $m_{\mathcal{C}}$ amount to saying that the concatenation operation is compatible with composition of morphisms - a condition that can be easily checked.

For any functor $F : \mathcal{C} \to \mathcal{D}$ the following diagram commutes strictly

$$\begin{tikzcd} MM\mathcal{C} \arrow[r, "m_{\mathcal{C}}"] \arrow[d, "MMF"'] & M\mathcal{C} \arrow[d, "MF"] \\ MM\mathcal{D} \arrow[r, "m_{\mathcal{D}}"'] & M\mathcal{D} \end{tikzcd}$$

This makes m a strict 2-natural transformation.

The unit of the monad is given by the collection $u_{\mathcal{C}} : \mathcal{C} \to M\mathcal{C}$. It sends $X \xrightarrow{f} Y \in \mathcal{C}$ to $(X) \xrightarrow{(f)} (Y) \in M\mathcal{C}$. One can easily check that it too is a strict 2-natural transformation.

All the modifications in the definition of M are equalities. For example, it is clear that

$$=_{\mathcal{C}} : m_{\mathcal{C}} \cdot Mm_{\mathcal{C}} \to m_{\mathcal{C}} \cdot m_{M\mathcal{C}}$$

is an equality because the order of removal of inner brackets of a list $l \in M^3\mathcal{C}$ does not affect the final outcome. An isomorphic argument shows that the other two modifications are also equalities.

$M = (M, n, u, =, =, =)$ is called the *monoidal category monad*.

Example 5.2. The *symmetric monoidal category monad*, $S : \mathsf{Cat} \to \mathsf{Cat}$ sends each category to the free symmetric monoidal category on it (cf. Example 4.7). The multiplication $m : S^2 \to S$ again removed the inner bracket between objects, but some care needs to be taken with regards to the morphisms.

For any category $\mathcal{C}$, the category $S\mathcal{C}$ contains two types of morphisms. The first type is just a list of morphisms of $\mathcal{C}$. The second type consists of symmetries introduced by the free construction. Let $f : l_1 \otimes l_2 \otimes \ldots \otimes l_n \to l'_1 \otimes l'_2 \otimes \ldots \otimes l'_n \in SS\mathcal{C}$ be a morphism. If $f = f_1 \otimes f_2 \otimes \ldots \otimes f_n$, with each f_i a morphism of $S\mathcal{C}$, set $m_{\mathcal{C}}(f)$ to be a list of morphisms obtained by removing all the inner brackets of f. f might also be a symmetry freely introduced in the construction of $SS\mathcal{C}$ from $S\mathcal{C}$. Then the action of $m_{\mathcal{C}}$ os a two step process: first the mapping of Equation 3.2 is applied to the symmetry f, and then the result is embedded into $S\mathcal{C}\big(m_{\mathcal{C}}(l_1 \otimes l_2 \otimes \ldots \otimes l_n), m_{\mathcal{C}}({}'_1 \otimes l'_2 \otimes \ldots \otimes l'_n)\big)$. Diagram 3.4 shows exactly how this process works.

The unit is defined as for the monoidal category monad (Example 5.1). All the modifications are equalities, making S a strict monad.

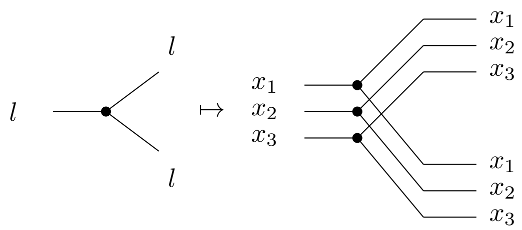
Example 5.3. The *finite product category monad*, $F : \mathsf{Cat} \to \mathsf{Cat}$, sends each category X to the free category with finite products on X (cf. Example 4.8). As in the previous example, $FF\mathcal{C}$ contains two types of morphisms: formal products of $F\mathcal{C}$ morphisms and new structural morphisms introduced by applying F to $F\mathcal{C}$. $m_{\mathcal{C}}$ removed the inner bracket (if they exist) of formal products of $F\mathcal{C}$-morphisms. I.e.

$$\Delta \times (f \times g) \times \delta \overset{m_{\mathcal{C}}}{\mapsto} \Delta \times f \times g \times \delta.$$

We illustrate $m_{\mathcal{C}}$'s actions on finite-product structure of $FF\mathcal{C}$ with an example. Let $l = (x_1 \times x_2 \times x_3)$ be a single-element list in $FF\mathcal{C}$. Then there there exists a morphism

$$\Delta : l \to l \times l \in FF\mathcal{C}$$

then $m_{\mathcal{C}}$'s of Δ action can be visualised as follows:



The unit sends x to the single-elements list $(x) \in F\mathcal{C}$.

This monad is also strict.

Example 5.4. Recall the presheaf functor P of Example 4.10. Following Section 7 in [Curien, 2008], we restrict the codomain of P to Cat obtaining a new functor to which we shall refer again as the presheaf functor. This new P can made into a monad. The unit of P is given by the Yoneda embedding

$$u_C : \mathcal{C} \to P\mathcal{C}, \quad X \mapsto \mathcal{C}(-, X).$$

Its multiplication, $m : P^2 \to P$, is given by the following left Kan extension

$$\begin{tikzcd} P\mathcal{C} \arrow[rr, "u_{P\mathcal{C}}"] \arrow[dr, "1_{P\mathcal{C}}"'] & & PP\mathcal{C} \arrow[dl, "m_{\mathcal{C}}"] \\ & P\mathcal{C} & \end{tikzcd}$$

which can be translated to an explicit formula for any $\mathcal{F} : (P\mathcal{C})^{\mathrm{op}} \to \mathsf{Set}$

$$\begin{aligned} m_{\mathcal{C}}(\mathcal{F}) &= \int^{\mathcal{F} \in P\mathcal{C}} F \times PP\mathcal{C}(\mathcal{F}, u_{P\mathcal{C}}(F)) \\ &\simeq \int^{F \in P\mathcal{C}} F \times \mathcal{F}(F). \end{aligned}$$

P is not a strict 2-monad. For example, the isomorphism $\lambda_{\mathcal{C}}$ is derived from the density formula for presheafs:

$$m_{\mathcal{C}}(P\mathcal{C}(-, \mathcal{G})(X) = \int^{f \in P\mathcal{C}} P\mathcal{C}(f, \mathcal{G}) \times f(X) \simeq \mathcal{G}(X).$$

Example 5.5. It was shown in Corollary 3.3 of [Cheng et al., 2004] that any 2-monad P induces a psudo-adjunction

$$(\tilde{-}) \dashv G : \mathsf{kl}(P) \to \mathcal{K}$$

One can show that the right adjoint, G, is given by

- On 0-cells $GX = PX$.
- on 1-cells and 2-cells by

$$G\Big(\begin{tikzcd} X \arrow[r, bend left, "\varphi"] \arrow[r, bend right, "\psi"'] \arrow[r, phantom, "\Downarrow\xi"] & Y \end{tikzcd} \Big) = \begin{tikzcd} PX \arrow[r, bend left, "P\varphi"] \arrow[r, bend right, "P\psi"'] \arrow[r, phantom, "\Downarrow\xi"] & P^2Y \arrow[r, "m_Y"] & PY \end{tikzcd} ;$$

The unit of this adjunction is given by

$$\eta_X : X \to PX = GX_* = X \xrightarrow{u_X} PX \in \mathcal{K}$$

and the counit is

$$\epsilon_X : PX \nrightarrow X = 1_{PX} : PX \nrightarrow X^1.$$

The adjunction laws translate to the following equations

$$X \overset{(u_X)_*}{\nrightarrow} PX \overset{1_{PX}}{\nrightarrow} X \simeq X \overset{u_X}{\nrightarrow} X \tag{5.1}$$

[1]We abuse notation and write 1_{PX} both for the functor $PX \to PX$ and for the profunctor $PX \nrightarrow X$

and

$$PX \xrightarrow{u_{PX}} P^2X \xrightarrow{P(1_{PX})} P^2X \xrightarrow{m_X} PX \;\simeq\; PX \xrightarrow{1_{PX}} PX \tag{5.2}$$

with additional coherence conditions which we ignore at this point.

5.2 2-monad morphisms

Let $S = (S, m, u, \tau, \lambda, \rho)$ and $(T, m, u, \tau', \lambda', \rho')$ be two 2-monads on a bicategory $\mathcal{K}$. A *2-monad morphism* is a triple $(\alpha, \alpha_m, \alpha_u)$ where $\alpha : S \to T$ is a pseudo-natural transformation and α_m and α_u are invertible modifications

$$\begin{tikzcd}
S^2 \arrow[r, "S\alpha"] \arrow[d, "m"'] & ST \arrow[r, "\alpha T"] \arrow[d, phantom, "\Downarrow\alpha_m"] & T^2 \arrow[d, "m'"] \\
S \arrow[rr, "\alpha"'] & & S
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
1_{\mathcal{K}} \arrow[r, "u"] \arrow[dr, "u'"'] & S \arrow[d, "\alpha"] \arrow[dl, phantom, "\Downarrow\alpha_u"] \\
& T
\end{tikzcd}$$

such that the following coherence conditions hold:

$$\begin{tikzcd}
& S^3 \arrow[r, "S^2\alpha"] \arrow[dl, "mS"'] \arrow[d, "Sm"] & S^2T \arrow[r, "S\alpha T"] \arrow[d, phantom, "\Downarrow S\alpha_m"] & ST^2 \arrow[r, "\alpha T^2"] \arrow[d, "Sm"'] \arrow[dr, phantom, "\simeq"] & T^3 \arrow[d, "Tm"] \\
S^2 \arrow[r, phantom, "\Downarrow\tau"] \arrow[dr, "m"'] & S^2 \arrow[rr, "S\alpha"] \arrow[d, "m"] & \arrow[d, phantom, "\Downarrow\alpha_m"] & ST \arrow[r, "\alpha T"] & T^2 \arrow[d, "m"] \\
& S \arrow[rrr, "\alpha"'] & & & T
\end{tikzcd}
=
\begin{tikzcd}
S^3 \arrow[r, "S^2\alpha"] \arrow[d, "mS"'] \arrow[dr, phantom, "\simeq"] & S^2T \arrow[r, "S\alpha T"] \arrow[d, "mT"] & ST^2 \arrow[r, "\alpha T^2"] \arrow[d, phantom, "\Downarrow\alpha_m T"] & T^3 \arrow[d, "mT"'] \arrow[dr, "Tm"] & \\
S^2 \arrow[r, "S\alpha"] \arrow[d, "m"] & TS \arrow[rr, "\alpha T"] & \arrow[d, phantom, "\Downarrow\alpha_m"] & T^2 \arrow[r, phantom, "\Downarrow\tau'"] \arrow[d, "m"'] & T^2 \arrow[dl, "m"] \\
S \arrow[rrr, "\alpha"'] & & & T &
\end{tikzcd}$$

$$\begin{tikzcd}
& S^2 \arrow[r, "S\alpha"] \arrow[d, "m"] & ST \arrow[r, "\alpha T"] \arrow[d, phantom, "\Downarrow\alpha_m"] & T^2 \arrow[d, "m"] \\
S \arrow[ur, "Su"] \arrow[r, "\mathrm{id}"'] \arrow[ur, phantom, "\Downarrow\rho"'] & S \arrow[rr, "\alpha"'] & & T
\end{tikzcd}
\qquad
\begin{tikzcd}
S^2 \arrow[r, "S\alpha"] \arrow[dr, phantom, "\Downarrow S\alpha_u"] & ST \arrow[r, "\alpha T"] \arrow[d, phantom, "\simeq"] & T^2 \arrow[dr, "m"] & \\
S \arrow[u, "Su"] \arrow[ur, "Su"'] \arrow[rr, "\alpha"'] & & T \arrow[u, "Tu"] \arrow[u, phantom, "\Downarrow\rho'"', shift right=3] \arrow[r, "\mathrm{id}"'] & T
\end{tikzcd}$$

and

$$\begin{tikzcd}
& S^2 \arrow[r, "S\alpha"] \arrow[d, "m"] & ST \arrow[r, "\alpha T"] \arrow[d, phantom, "\Downarrow\alpha_m"] & T^2 \arrow[d, "m"] \\
S \arrow[ur, "uS"] \arrow[r, "\mathrm{id}"'] \arrow[ur, phantom, "\Downarrow\lambda"'] & S \arrow[rr, "\alpha"'] & & T
\end{tikzcd}
\qquad
\begin{tikzcd}
S^2 \arrow[r, "S\alpha"] & ST \arrow[r, "\alpha T"] \arrow[d, phantom, "\simeq"] & T^2 \arrow[dr, "m"] & \\
S \arrow[u, "uS"] \arrow[rr, "\alpha"'] & & T \arrow[ul, "uT"] \arrow[ul, phantom, "\Uparrow\alpha_u T"'] \arrow[u, "uT"'] \arrow[u, phantom, "\Downarrow\lambda'", shift left=3] \arrow[r, "\mathrm{id}"'] & T
\end{tikzcd}$$

Example 5.6. Consider the monoidal category monad $M : \mathsf{Cat} \to \mathsf{Cat}$ and the symmetric monoidal category monad $S : \mathsf{Cat} \to \mathsf{Cat}$ of the previous section. There is a monad morphism

$$i : M \to S$$

given by setting, for every category $\mathcal{C}$, the functor $i_{\mathcal{C}}$ to be the inclusion $M\mathcal{C} \hookrightarrow S\mathcal{C}$.

We verify the existence of the required modifications. The diagram

$$\begin{tikzcd} M^2 \arrow[r, "Mi"] \arrow[d, "m"'] & MS \arrow[r, "iS"] & S^2 \arrow[d, "m'"] \\ M \arrow[rr, "i"'] & & S \end{tikzcd}$$

strictly commutes since the process of removing inner brackets commutes with set inclusion.

The commutativity of

$$\begin{tikzcd} 1_{\mathsf{Cat}} \arrow[r, "u"] \arrow[dr, "u'"'] & M \arrow[d, "i"] \\ & S \end{tikzcd}$$

is also clear.

Example 5.7. The inclusions of symmetric monoidal categories into finite product categories gives rise to a monad morphism. The argument is identical to the one given in Example 5.6.

5.3 2-algebras

As with single-dimensional monads, 2-monads also have algebras. An algebra for a 2-monad is defined similarly to an algebra for a monad, expect that the equalities in Equations 2.2 are replaced by invertible modifications that are subject to coherence conditions. A similar story repeats for morphisms of algebras. This data forms a bicategory. We refer the reader to Section 2.4 of [Tanaka, 2005] for a detailed definition.

We shall often talk about monads whose bicategory of algebras consists particular categories with structure. For example, the bicategory of algebras for the monoid monad of Example 5.1 consist of all monoidal categories and monoidal functors. Similarly, the 2-categpory of all symmetric categories and symmetric monoidal functors is the category of algebras for the monad S of Example 5.2.

5.4 Kleisli bicategories

Let $P : \mathcal{K} \to \mathcal{K}$ be a 2-monad on a bicategory $\mathcal{K}$. The *Kleisli bicategory of P*, written kl$P$ is defined as follows. Its objects are just the objects of $\mathcal{K}$:

$$\mathrm{ob}(\mathsf{kl}(P)) = \mathrm{ob}\mathcal{K}$$

For any two $X, Y \in \mathrm{ob}(\mathsf{kl}(P))$ we put

$$\mathsf{kl}(P)(X, Y) = \mathcal{K}(X, PY),$$

with composition

$$\circ_{X,Y,Z} : \mathcal{K}(X, PY) \times \mathcal{K}(Y, PZ) \to \mathcal{K}(X, PZ)$$

given by

$$(X \xrightarrow{\varphi} PY) \circ (Y \xrightarrow{\psi} PZ) = X \xrightarrow{\varphi} PY \xrightarrow{P\psi} PPZ \xrightarrow{m_Z} PZ \,.$$

It is clear that this composition is a functor. The unit is given by the unit of the monad P.

Given any $\varphi : X \to Y$ in $\mathsf{kl}(P)$, the isomorphism $\varphi \circ 1_X \simeq \varphi$ is given by

$$\begin{tikzcd}
X \arrow[r, "u_X"] \arrow[dr, "\varphi"'] & PX \arrow[r, "P\varphi"] \arrow[d, phantom, "\Downarrow"] & P^2Y \arrow[r, "m_Y"] \arrow[d, phantom, "\Downarrow"] & PY \\
& PY \arrow[ur, "u_{PY}"] \arrow[urr, bend right, "1_{PY}"'] & &
\end{tikzcd}$$

the isomorphism $1_Y \circ \varphi \simeq \varphi$ is given by

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & PY \arrow[r, "Pu_Y"] \arrow[dr, bend right, "1_{PY}"'] \arrow[dr, phantom, "\Downarrow"] & P^2Y \arrow[d, "m_Y"] \\
& & PY
\end{tikzcd}$$

and with $\zeta : Z \to PW$, $\zeta \circ (\psi \circ \varphi) \simeq (\zeta \circ \psi) \circ \varphi$ is given by

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & PY \arrow[r, "P\psi"] & P^2Z \arrow[r, "m_Z"] \arrow[dr, bend right, "P^2\zeta"'] & PZ \arrow[r, "P\zeta"] \arrow[d, phantom, "\Downarrow"] & P^2W \arrow[r, "m_W"] \arrow[d, phantom, "\Downarrow"] & PW \\
& & & P^3W \arrow[ur, "m_{PW}"] \arrow[r, "Pm_Z"'] & P^2W \arrow[ur, bend right, "m_W"'] &
\end{tikzcd}$$

Example 5.8. We investigate the Kleisli bicategory of the Presheaf monad (Example 5.4). The objects of $\mathsf{kl}(P)$ are categories. The morphisms are functors

$$X \to PY = \mathsf{Set}^{Y^{\mathrm{op}}}$$

which, since Cat is Cartesian closed, are equivalent to functors

$$Y^{\mathrm{op}} \times X \to \mathsf{Set};$$

explicitly, the equivalence is given by

$$\varphi(x)(y = \varphi(y)(x)$$

Let $X \xrightarrow{\varphi} Y \xrightarrow{\psi} Z$ be morphisms in $\mathsf{kl}(P)$. Their composition is

$$\begin{aligned}
\psi \cdot \varphi(x)(z) &= m_Z \cdot P\psi \cdot \phi(x)(z) \\
&= m_Z\big(P\psi(\varphi(x))\big)(y) \\
&\simeq m_Z\Big(\int^{y \in Y} PZ(-, \psi(y)) \times \varphi(x)(y)\Big)(z) \\
&\simeq \int^{G \in PZ} \int^{y \in Y} PZ(G, \psi(y)) \times \varphi(x)(y) \times G(z) \\
&\simeq \int^{y \in Y} \varphi(x)(y) \times \int^{G \in PZ} G(z) \times PZ(G, \psi(y)) \\
&\simeq \int^{y \in Y} \varphi(x)(y) \times \psi(y)(z).
\end{aligned}$$

The composition we computed above is exactly the composition of ψ with ϕ when we view them as profunctors via the above identification. From this we conclude that the categories $\mathsf{kl}(P)$ and Prof (see Example 4.5) are biequivalent.

5.5 2-distributive laws

Given two 2-monads, S and T, we can compose their underlying 2-functors. A 2-distributive law allows us to create a monad structure on the composite 2-functor ST. 2-distributive laws are substantially more complicated than their 1-dimensional cousins. Because of this, we shall study them in stages. We begin with a discussion a distributive law of a 2-functor over a monad. The full definition of a 2-distributive law is given only after we understood this simpler scenario in some detail.

5.5.1 Distributing functors over monads

Let (P, m, u) be a monad on a 2-category $\mathcal{K}$ and let S be a 2-functor on $\mathcal{K}$. A *2-distributive law* (also pseudodistributive law) of S over P is a triple $(\delta, \delta_m, \delta_u)$ where

$$\delta : SP \to PS$$

is a 2-natural transformation and

$$\begin{tikzcd}
SP^2 \arrow[r, "\delta P"] \arrow[d, "Sm"'] & PSP \arrow[r, "P\delta"] \arrow[d, phantom, "\Downarrow\delta_m"] & P^2S \arrow[d, "mS"] \\
SP \arrow[rr, "\delta"'] & & PS
\end{tikzcd}
\qquad
\begin{tikzcd}
S \arrow[d, "Su"'] \arrow[dr, "uS"] \arrow[d, phantom, shift left=3, "\Downarrow\delta_u"] & \\
SP \arrow[r, "\delta"'] & PS
\end{tikzcd}$$

are invertible modification subject to the following three coherence conditions

$$
\begin{tikzcd}
 & SP^3 \arrow[dl, "SPm"'] \arrow[d, "SmP"] \arrow[r, "\delta P^2"] & PSP^2 \arrow[r, "P\delta P"] \arrow[d, phantom, "\Downarrow\delta_m P"] & P^2SP \arrow[r, "P^2\delta"] \arrow[d, "mSP"'] \arrow[dr, phantom, "\text{nat.}"] & P^3S \arrow[d, "mPS"] \\
SP^2 \arrow[dr, "Sm"'] \arrow[r, phantom, "S\tau"] & SP^2 \arrow[rr, "\delta P"] \arrow[d, "Sm"] & \arrow[d, phantom, "\Downarrow\delta_m"] & PSP \arrow[r, "P\delta"] & P^2S \arrow[d, "mS"] \\
 & SP \arrow[rrr, "\delta"'] & & & PS
\end{tikzcd}
=
\begin{tikzcd}
SP^3 \arrow[r, "\delta P^2"] \arrow[d, "SPm"'] \arrow[dr, phantom, "\text{nat.}"] & PSP^2 \arrow[r, "P\delta P"] \arrow[d, "PSm"] & P^2SP \arrow[r, "P^2\delta"] \arrow[d, phantom, "\Downarrow\delta_m"] & P^3S \arrow[d, "PmS"'] \arrow[dr, "mPS"] & \\
SP^2 \arrow[r, "\delta P"] \arrow[d, "Sm"'] & PSP \arrow[rr, "P\delta"] \arrow[d, phantom, "\Downarrow\delta_m"] & & P^2S \arrow[d, "mS"'] \arrow[r, phantom, "\Downarrow\tau S"] & P^2S \arrow[dl, "mS"] \\
SP \arrow[rrr, "\delta"'] & & & PS &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & SP^2 \arrow[r, "\delta P"] \arrow[d, "Sm"] \arrow[dl, phantom, "\Downarrow S\rho"] & PSP \arrow[r, "P\delta"] \arrow[d, phantom, "\Downarrow\delta_m"] & P^2S \arrow[d, "mS"] \\
SP \arrow[ur, "SuP"] \arrow[r, "\mathsf{id}"'] & SP \arrow[rr, "\delta"'] & & PS
\end{tikzcd}
\qquad
\begin{tikzcd}
SP^2 \arrow[r, "\delta P"] \arrow[dr, phantom, "\Downarrow\delta_u P"] & PSP \arrow[r, "P\delta"] \arrow[d, phantom, "\text{nat.}"] & P^2S \arrow[dr, "mS"] \arrow[d, phantom, "\Downarrow\rho S"] & \\
SP \arrow[u, "SuP"] \arrow[ur, "uSP"'] \arrow[rr, "\delta"'] & & PS \arrow[u, "uPS"] \arrow[r, "\mathsf{id}"'] & PS
\end{tikzcd}
$$

and

$$
\begin{tikzcd}
 & SP^2 \arrow[r, "\delta P"] \arrow[d, "Sm"] \arrow[dl, phantom, "\Downarrow S\lambda"] & PSP \arrow[r, "P\delta"] \arrow[d, phantom, "\Downarrow\delta_m"] & P^2S \arrow[d, "mS"] \\
SP \arrow[ur, "SPu"] \arrow[r, "\mathsf{id}"'] & SP \arrow[rr, "\delta"'] & & PS
\end{tikzcd}
\qquad
\begin{tikzcd}
SP^2 \arrow[r, "\delta P"] \arrow[dr, phantom, "\text{nat.}"] & PSP \arrow[r, "P\delta"] \arrow[dr, phantom, "\Uparrow P\delta_u"] & P^2S \arrow[dr, "mS"] \arrow[d, phantom, "\Downarrow\lambda S"] & \\
SP \arrow[u, "SPu"] \arrow[rr, "\delta"'] & & PS \arrow[ul, "PSu"] \arrow[u, "PuS"'] \arrow[r, "\mathsf{id}"'] & PS
\end{tikzcd}
$$

Example 5.9. The free monoidal category functor of Example 4.6 distributes over the presheaf monad of Example 5.4. For any category X the functors

$$\lambda_X : MPX \to PMX, \quad f_1 \otimes \ldots \otimes f_n \mapsto f_1(-) \times \ldots \times f_n(-)$$

defines a 2-distributive law.

Distributive laws allow us to extend the endofunctor S to the Kleisli category of P. More precisely, we call the 2-functor $\tilde{S}$ as *extension of a 2-functor S* when

$$
\begin{tikzcd}
\mathsf{kl}(P) \arrow[r, "\tilde{S}"] & \mathsf{kl}(P) \\
\mathcal{K} \arrow[u, "(-)_*"] \arrow[r, "S"'] & \mathcal{K} \arrow[u, "(-)_*"']
\end{tikzcd}
$$

commutes.

The relationship between extensions and distributive laws is captured by the following theorem
Theorem 5.10. *Given a 2-monad P a 2-functor S on a 2-category $\mathcal{K}$ the following a equivalent:*

1. *a 2-distributive law $(\lambda, \lambda_m, \lambda_u) : SP \to PS$*
2. *an extension $\tilde{S}$ of S to $\mathsf{kl}(P)$.*

The proof can be found in [Cheng et al., 2004]; we provide a sketch of the proof in Theorem 7.4.

The extension $\tilde{S}$ can be easily described. Since its an extension, it equals S on 0-cells. A 1-cell $\varphi : X \to Y$ in $\mathsf{kl}(P)$ is the same as a 1-cell $\varphi : X \to PY$ in $\mathcal{K}$. Set

$$\tilde{S}(\varphi) = \; SX \xrightarrow{S\varphi} SPY \xrightarrow{\delta_Y} PSY$$

which is a 1-cell from SX to SY in $\mathsf{kl}(P)$.

Example 5.11. The distributive law of Example 5.9 gives rise to a 2-functor $\tilde{M} : \mathsf{Prof} \to \mathsf{Prof}$ that equals M on categories and is defined by

$$\tilde{M}(\varphi)(y_1 \otimes \ldots \otimes y_m, x_1 \otimes \ldots \otimes x_n) = \begin{cases} \varphi(y_1, x_1) \times \ldots \times \varphi(y_n, x_n), \text{ if } n = m; \\ \emptyset, \text{ else,} \end{cases}$$

for any profunctor $\varphi : X \nrightarrow Y$.

One can also define a distributive law of *a monad P over a functor S*. This definition is identical to ours, with the roles of P and S reversed. That is, it is given by a 2-natural transformation

$$\lambda : PS \to SP.$$

and two modifications that satisfy three coherence conditions. Full details can be found in Chapter 5 of [Tanaka, 2005].

5.5.2 Distributing monads over monads

Let (S, m, u) and (P, μ, η) be monad on a 2-category $\mathcal{K}$. A *2-distributive law of S over P* is a a 5-tuple $(\lambda, \lambda_m, \lambda_u, \lambda_\mu, \lambda_\eta)$ where

$$\lambda : SP \to PS$$

is a 2-natural transformation and λ_μ, λ_η, λ_m and λ_u, are invertible modifications

$$\begin{tikzcd} SP^2 \arrow[r, "\lambda P"] \arrow[d, "S\mu"'] & PSP \arrow[r, "P\lambda"] \arrow[d, phantom, "\Downarrow\lambda_\mu"] & P^2S \arrow[d, "\mu S"] \\ SP \arrow[rr, "\lambda"'] & & PS \end{tikzcd} \qquad \begin{tikzcd} S \arrow[d, "S\eta"'] \arrow[dr, "\eta S"] \arrow[d, phantom, shift left=3ex, "\Downarrow\lambda_\eta"] & \\ SP \arrow[r, "\lambda"'] & PS \end{tikzcd}$$

$$\begin{tikzcd} S^2P \arrow[r, "S\lambda"] \arrow[d, "mP"'] & SPS \arrow[r, "\lambda S"] \arrow[d, phantom, "\Downarrow\lambda_m"] & PS^2 \arrow[d, "Pm"] \\ SP \arrow[rr, "\lambda"'] & & PS \end{tikzcd} \qquad \begin{tikzcd} P \arrow[d, "uP"'] \arrow[dr, "Pu"] \arrow[d, phantom, shift left=3ex, "\Downarrow\lambda_u"] & \\ SP \arrow[r, "\lambda"'] & PS \end{tikzcd}$$

subject to ten coherence condition given in Chapter 7 of [Tanaka, 2005].

It took some time to precisely identify the coherence condition for 2-distributive laws of monads over monads. The first attempt was made in [Marmolejo, 1999] where nine coherence conditions we presented. In [Tanaka, 2005] Tanaka added an addition coherence condition to the list. But three years later, [Marmolejo and Wood, 2008] showed that Tanaka's condition is in fact derivable from that original nine. Interestingly, the same work showed that one of the nine conditions is derivable from the other eight. Thus the minimal definition for a 2-distributive involved eight coherence conditions; those can be found in the original paper [Marmolejo and Wood, 2008] or in the survey [Gambino, 2009].

Example 5.12. The 2-distributive law of Example 5.9 is actually a 2-distributive law of the 2-monad M (cf. 5.1) over the 2-monad P (cf. Example 5.4).

The formula

$$f_1 \otimes \ldots \otimes f_n \mapsto f_1(-) \times \ldots \times f_n(-)$$

follows from abstract considerations. Let $(X, \otimes, I)$ be a monoidal category. Then there is a monoidal tensor product on PX given by the *Day convolution* which, given profunctors $F, G : X^{\mathrm{op}} \to \mathsf{Set}$, is defined by

$$F * G = \int^{x,y \in X} Fx \times Gy \times X(-, x \otimes y).$$

Is is well know that this tensor product along with unit $J = \eta_X(I)$, makes $(PX, *, J)$ a monoidal closed category.

Since MX is the free monoidal category on X, it possesses the the following universal property: for any strict monoidal category Y, and a functor $H : X \to Y$, there is a unique strict monoidal functor $\hat{H}$ that makes

$$\begin{tikzcd}
X \arrow[r, "u_X"] \arrow[dr, "H"'] & MX \arrow[d, dashed, "\hat{H}"] \\
& Y
\end{tikzcd}$$

commute. In particular, consider the situation

$$\begin{tikzcd}
PX \arrow[r, "u_{PX}"] \arrow[dr, "Pu_X"'] & MPX \arrow[d, dashed, "\lambda_X"] \\
& PMX
\end{tikzcd}$$

We can now compute the formula for λ_X. For the sake of clarity, we consider the action of λ_X on only two presheaves f and g. We write $[x]$ for the list $x_1 \otimes \ldots \otimes x_n$ and similarly write $[y]$ and $[z]$ for lists of objects of X.

$$\begin{aligned}
\lambda_X(f \otimes g)([x]) &= \lambda_X(u_{PX}(f) \otimes u_{PX}(g))([x]) \\
&\simeq \big(Pu_X(f) * Pu_X(g)\big)([x]) \\
&\simeq \int^{[y],[z] \in MX} Pu_X(f)([y]) \times Pu_X(g)([z]) \times MX([x], [y] \otimes [z]) \\
&\simeq \int^{[y],[z]} \Big(\int^{a \in X} f(a) \times MX([y], a) \times \int^{b \in X} g(b) \times MX([z], b) \Big) \times MX([x], [y] \otimes [z]) \\
&\simeq \int^{a,b \in X} f(a) \times g(b) \times \Big(\int^{[y],[z]} MX([x], [y] \otimes [z]) \times MX([y], a) \times MX([z], b) \Big) \\
&\simeq \int^{a,b \in X} f(a) \times g(b) \times MX([x], a \otimes b) \\
&= \begin{cases} f(x_1) \times g(x_2), \text{ if } x = x_1 \otimes x_2; \\ \emptyset, \text{ else,} \end{cases}
\end{aligned}$$

The last isomorphism holds because $MX([y], a) \times MX([z], b) \simeq MX([y] \otimes [z], a \otimes b)$ and the coend is taken over $[y] \otimes [z]$

Example 5.13. The Symmetric Monoidal Category monad S of Example 5.2 and the Finite Product Category monad F of Example 5.3 also distribute over Presheaf monad P. The 2-distributive laws are almost identical to the one in the previous example. We record then here for completeness. We define $\rho : SP \to PS$ by

$$\rho_X\big(f_1 \otimes \ldots \otimes f_n\big)(x_1 \otimes \ldots \otimes x_k) = \int^{x'_1,\ldots,x'_n \in X} f_1(x'_1) \times \ldots \times f_n(x'_n) \times SX(x_1 \otimes \ldots \otimes x_k, x'_1 \otimes \ldots \otimes x'_n).$$

and $\xi : FP \to PF$ by

$$\xi_X\big(f_1 \otimes \ldots \otimes f_n\big)(x_1 \otimes \ldots \otimes x_k) = \int^{x'_1,\ldots,x'_n \in X} f_1(x'_1) \times \ldots \times f_n(x'_n) \times FX(x_1 \otimes \ldots \otimes x_k, x'_1 \otimes \ldots \otimes x'_n).$$

The proof that the formulas we gave above are indeed 2-distributive laws can be found in [Curien, 2008]. That paper elegantly considers all there monads M, S and F at the same time and so provides a single proof that works for all three constructions.

Let (S, m, u) and P be monads on $\mathcal{K}$. An extension $\tilde{S} = (\tilde{S}, \tilde{m}, \tilde{u})$ of S to $\mathsf{kl}(P)$ is a monad the 2-functor, the 2-natural transformations and the modifications are all extensions of the original data. (See Chapter **??** for precise definitions).

The following important Theorem relates liftings and extensions of monads with 2-distributive laws.

Theorem 5.14. *Let S and P be 2-monads on a 2-category $\mathcal{K}$. Then the following are equivalent:*

1. *a 2-distributive law $\lambda : SP \to PS$;*
2. *an extension $\tilde{S}$ of S to the Kleisli bicategory of P;*
3. *a lifting of P to the bicategory of P-algebras.*

See [Cheng et al., 2004] for a proof. A detailed discussion of (1) $\iff$ (3) can be found in [Tanaka, 2005].

Example 5.15. Examples 5.12 and 5.13 describe three distributive laws of monads over the presheaf monad P. Hence by Theorem 5.14 the Monoidal Categories Monad, the Symmetric Monoidal Categories monad and the Finite Product Categories monad can all be extended to $\mathsf{kl}(P)$, i.e. to Prof..

Consequently, we obtain three 2-functors $\tilde{M}$, $\tilde{S}$ and $\tilde{F}$ on the bicategory Prof. Let $T \in \{M, S, F\}$ be one of our 2-monads and $\delta_T \in \lambda, \rho, \xi$ be a corresponding 2-distributive law. The actions of the extension $\tilde{T}$ on a profunctor $\varphi : X \to PY$ is given by

$$TX \xrightarrow{T\varphi} TPY \xrightarrow{\delta_T} PTY$$

This allows us to derive explicit formulas for the tree extensions. Let $\varphi : X \nrightarrow Y$ be a profunctor then

- $$\tilde{M}\varphi(y_1 \otimes \ldots \otimes y_n, x_1 \otimes \ldots \otimes x_n) = \varphi(y_1, x_1) \times \ldots \times \varphi(y_n, x_n)$$

- $$\tilde{S}\varphi(y_1 \otimes \ldots \otimes y_n, x_1 \otimes \ldots \otimes x_n) = \int^{y'_1,\ldots,y'_n \in Y} \varphi(y'_1, x_1) \times \ldots \times \varphi(y'_n, x_n) \times SX(y_1 \otimes \ldots \otimes y_n, y'_1 \otimes \ldots \otimes y'_n)$$

- $$\tilde{F}\varphi(y_1\otimes\ldots\otimes y_p, x_1\otimes\ldots\otimes x_n) = \int^{{y'}_1,\ldots,{y'}_n\in Y} \varphi(y'_1, x_1)\times\ldots\times\varphi(y'_n, x_n)\times PX(y_1\otimes\ldots\otimes y_p, {y'}_1\otimes\ldots\otimes {y'}_n)$$

In the next Chapter we shall see that we monads in the Kleisli bicategories of $\tilde{M}$, $\tilde{S}$ and $\tilde{F}$ are operads, symmetric operads and Lawvere theories, respectively.

Chapter 6

Internal Category Theory

The aim of this chapter is to show how the 2-categorical tools we developed in Chapters 4 and 5 can be used to define a setting that makes it easy to formulate and combine various gadgets from categorical universal algebra. The monad again plays a central role. But instead of studying monads on a 2-category $\mathcal{K}$, we now study monads *inside* $\mathcal{K}$. Monads in different $\mathcal{K}$ correspond to various familiar constructions.

In the first example we show that a category is a monad in the bicategory of Spans (cf. [Rosebrugh and Wood, 2002]). Next we study monads in Prof. There internal monads add a new categorical structure to a category X in a way that interacts nicely with the existing structure of X. We follow [Lack, 2004] and show that this view allows us to define PROs, PROPs and Lawvere theories as monads in a bicategory Prof(Mon) of internal profunctors in Mon.

Operadic theories are more difficult to define because their tree-like composition does not naturality fit into a simple categorical framework. [Hyland, 2010] showed that operatic composition can be recovered by working in certain Kleisli bicategories. We focus on three particularly important examples. The first one studies monads in kl($\tilde{M}$), where $\tilde{M}$ is the extension of the monoidal category monad from Cat to Prof. We show that monads on 1 in that bicategory are precisely non-symmetric operads. Symmetric operads turn out to be monads on 1 in the Kleisli bicategory kl($\tilde{S}$), where S is the symmetric-monoidal category monad on Cat and $\tilde{S}$ is its extension to Prof. Finally, monads on 1 in kl($\tilde{F}$) are Cartesian operads. Here F is the finite product category monad on Cat.

We also give several examples of concrete theories. In particular, we describe the theory for commutative semi-groups in symmetric monoidal categories as a monad on 1 in kl($\tilde{S}$).

Distributive laws can also be internalised. Such internal distributive laws give a new method for composing structure. This ability is one of the primary reasons for representing familiar structures as monads inside bicategories. We explore the ability to compose categories, compose PROs, PROPs and Lawvere theories. We finish with a discussion of a concrete distributive law between the concrete operadic theories we defined earlier in the chapter.

6.1 Monads in 2-categories

Let $\mathcal{K}$ be a 2-category. A *monad in* $\mathcal{K}$ is a 1-cell $s : X \to X$ equipped with 2-cells $m : s^2 \to s$ and $u : 1_X \to s$ such that m and u satisfy the now familiar condition for a monad (cf. Section 2.2).

Example 6.1. Single-dimensional monads are monads in the 2-category Cat.

Example 6.2. A monad in $\mathsf{Span}(\mathsf{Set})$ consists of a set X, an endo-1-cell

$$\begin{tikzcd} X & & X \\ & E \arrow[ul, "d"] \arrow[ur, "c"'] & \end{tikzcd}$$

and 2-cells $\kappa : E \times_X E \to E$ and $\iota : X \to E$ that satisfy the associativity and unit axioms.

Let X be a set. The category $\mathsf{Mon}(\mathsf{Span}(\mathsf{Set}))(X, X)$ of monads on X in $\mathsf{Span}(\mathsf{Set})$ and monad morphisms between them is isomorphic to the subcategory Cat_X of Cat - viewed as a 1-category - consisting of all the categories $\mathcal{C}$ with $\mathrm{ob}\mathcal{C} = X$ and identity-on-objects functors between them. See Appendix B in [Akhvlediani, 2009] for a proof.

Example 6.3. We can represent strict monoidal categories as internal monads in a bicategory $\mathsf{Span}(\mathsf{Mon})$ of spans internal to the category Mon. Briefly, the category $\mathsf{Span}(\mathsf{Mon})$ is the same as Span, but the 0-cells are monoids and 1-cells are pairs of monoid homomorphisms. More detail can found in [Lack, 2004].

Example 6.4. Monads in Prof describe many interesting structures. First consider a monad

$$t : X \nrightarrow X$$

over a discrete category X. Since Span is equivalent to a full subcategory of Prof with 0-cells discrete categories, the monad t is equivalent to a category whose class of objects is X.

When X is not discrete, the monad $t : X \to X$ can again be thought of as a new category structure on X, but one that interacts nicely with the existing morphisms in X. We investigate this idea further in the discussion bellow.

Imagine that you wish to add new morphisms to an existing category X and that we wish the new morphisms to compose with each other and with the existing morphisms on X. The first task in this endeavour is to assign every pair $x, x' \in \mathrm{ob}X$ a new set of morphisms $t(x, x')$ which contains $X(x, x')$. We denote X-morphisms with English letters $f, g, h, \ldots$ and t-morphisms that are not in X with Greek letters $\varphi, \psi, \xi, \ldots$.

We wish to compose existing morphisms in X with the new morphisms from t. For this we need maps

$$t(x, x') \times X(x', x'') \to t(x, x'') \text{ and } X(x, x') \times t(x', x'') \to t(x, x''). \tag{6.1}$$

that respect the composition in X. Such maps are given by a functor $t : X^{\mathrm{op}} \times X \to \mathsf{Set}$. Indeed, one can verify the functor $t(x, -)$ gives the left map in Equation 6.1 and $t(-, x'')$ gives the right one. Let

$f : x \to x'$, $\varphi \in t(x', x'')$, and $g : x'' \to x'''$. Define

$$\varphi * f := t(f, x'')(\varphi) \in t(x, x'') \text{ and } g * \varphi := t(x', g)(\varphi) \in t(x', x''').$$

The functoriality of t also gives rise to the identities

$$f * g = f \cdot g \text{ and } 1_{x'} * f = f = f * 1_x$$

which indicate the compatibility of the operations $\cdot$ and $*$.

We determined that $t : X \nrightarrow X$ is a profunctor. We further suppose that it is a monad. Then the unit of the monad

$$X(x, x') \to t(x, x')$$

ensures that all the original morphisms of X are contained in the new morphism family t.

The composition of t-morphisms is given by the multiplication of the monad. So, for any pair of composible morphisms $\varphi : x \to y$ and $\psi : y \to x'$ in t, the multiplication produces the composite $\psi \Box \varphi$. The equivalence relation in profunctor composition gives rise to important identities for this multiplication. The multiplication of t takes the form

$$\int^{y \in X} t(x, y) \times t(y, x') \overset{\Box}{\to} t(x, x') \tag{6.2}$$

Let $\varphi : x \to y$, $\psi : y' \to x'$ and $f : y \to f'$ be given. Then there are two ways to compose this data. We can first form $\psi * f$ and then compose it with φ. This gives

$$(\psi * f)\Box\varphi$$

Or we can form $f * \varphi$ and then compose that with ψ:

$$\psi\Box(f * \varphi)$$

But since the pairs $(\varphi, (\psi * f))$ and $((f * \varphi), \psi)$ are equal in the coend of 6.2 they are mapped to the same composite by $\Box$.

As a consequence of the unit law for the monad t, for every $f : x \to x'$, $\varphi : x' \to x''$ and $g : x'' \to x'''$,

$$\varphi\Box f = \varphi * f \text{ and } g\Box\varphi = g * \varphi.$$

This hypothetical scenario is actually very useful in practical situations, in so far as these exist in Category Theory. Let $X = S1$ be the free symmetric monoidal category on a single object category. The only morphisms in X are the symmetries. A monad $t : X \nrightarrow X$ in Prof gives a way of adding additional morphisms to the symmetric monoidal "base". The new morphisms will seamlessly interact with the symmetry morphisms of X.

This approach is not perfect. The issue is that while t-morphisms compose with symmetries, they are oblivious to the monoidal structure of X. For example, viewed as a functor, $t : X^{\mathrm{op}} \times X \to \mathsf{Set}$ ignores the monoidal structure of X. To remedy this, we can take internal monads in the bicategory $\mathsf{Prof}(\mathsf{Mon})$ rather than in Prof. The former bicategory is called *the bicategory of internal profunctors in* Mon.

Basically, the idea is to make sure that X is a monoidal and that t preserves its monoidal structure. This approach was been explored in [Lack, 2004] where the reader can find further details.

The reliance of $\mathsf{Prof}(\mathsf{Mon})$ on Mon limits this setting. The problem is that there is no way to specify interaction of t-morphisms with the structure of X, if this structure is not monoidal. In the next section we present a framework that addresses this issue.

Example 6.5. In Example 6.4 we saw that monads in Prof and in $\mathsf{Prof}(\mathsf{Mon})$ are useful for appending additional morphisms to categories. And so the class of all monads $t : X \nrightarrow X$ models the class of categories that contains a subcategory X. Going back to Chapter 3, we see that PROs, PROPs, PROBs and Lawvere Theories can all be modelled as monads on suitable categories in $\mathsf{Prof}(\mathsf{Mon})$. [Lack, 2004] studies the implications of this for PROs and PROPs. [Akhvlediani, 2009] observes that PROBs can also be studied in that setting. [Akhvlediani, 2010] shows that Lawvere theories can also be viewed as monads and studies how these monads can be composed.

6.2 Monads in Kleisli bicategories

In in Example 5.15 of Chapter 5 we showed that the 2-monads M, S and F extend to the bicategory $\mathsf{kl}(P)$. This means that we can construct their Kleisli bicategories. In this section we study internal monads in these Kleisli bicategories. We shall see that these internal monads provide us with tools to model multicategories of various types. In particular, we shall see that plain, symmetric and Cartesian operads, as well as their coloured counterparts, can all be modelled elegantly in this setting.

6.2.1 Monads in $\mathsf{kl}(\tilde{M})$

We begin with the following important proposition.

Proposition 6.6 (Hyland)**.** *There is a one-to-one correspondence between monads*

$$t : 1 \nrightarrow 1$$

in $\mathsf{kl}(\tilde{M})$ *and plain operads.*

Sketch of proof. Lets begin with a plain operad A. We wish to construct a monad $t : 1 \nrightarrow 1$ in $\mathsf{kl}(\tilde{M})$. Define a functor

$$t : M1^{op} \times 1 \to \mathsf{Set}$$

by $t(n, 1) = A(n)$ for all $n \in \mathbb{N}$. Since 1 does not contain any non-trivial morphisms, this definition indeed gives a functor.

The unit of the monad is given by the natural transformation

$$u_1 : \mathbb{N}(1, 1) \to t(1, 1) \quad \mathsf{id}_1 \mapsto \mathsf{id}_A$$

To understand how to define the multiplication, we first compute the composite of t with itself in $\mathsf{kl}(\tilde{M})$:

$$t^2 = \ 1 \xrightarrow{t} M1 \xrightarrow{\tilde{M}t} MM1 \xrightarrow{\tilde{m}_1} M1$$

which evaluates to

$$
\begin{aligned}
m_1 \cdot \tilde{M}t \cdot t(n,1) &= \int^{k \in M1} t(k,1) \times \tilde{M}t \cdot \tilde{m}_1(n,k) \\
&= \int^{k \in M1} t(k,1) \times \int^{r_1 \otimes \ldots \otimes r_p \in MM1} \tilde{m}_1(n, r_1 \otimes \ldots \otimes r_p) \times \tilde{M}t(r_1 \otimes \ldots \otimes r_p, k) \\
&\simeq \int^{k \in M1} t(k,1) \times \int^{r_1 \otimes \ldots \otimes r_p \in MM1} M1(n, \sum_{i=1}^{p} r_i) \times \underbrace{t(r_1,1) \times \ldots \times t(r_p,1)}_{k \text{ times}} \\
&\simeq \int^{k \in M1} t(k,1) \times \int^{r_1 \otimes \ldots \otimes r_k \in MM1,\ \sum_{i=1}^{k} r_i = n} t(r_1,1) \times \ldots \times t(r_k,1) \\
&= \coprod_{k \in M1} t(k,1) \times \coprod_{\sum_{i=1}^{k} r_i = n} t(r_1,1) \times \ldots \times t(r_k,1)
\end{aligned}
$$

The composition of A gives us a passage

$$t(k) \times t(r_1,1) \times \ldots \times t(r_k,1) \to t(\sum_{i=0}^{k} r_i)$$

which gives us a morphism $t^2(n,1) \to t(n,1)$ by the universal property of coproducts. The associativity of multiplication and the unit law hold because of the corresponding axioms for A.

Conversely, given a monad t, we set $A(n) := t(n,1)$. Multiplication for A is then given by

$$A(k) \times A(r_1) \times \ldots \times A(r_k) \hookrightarrow \coprod_{k \in M1} A(k) \times \coprod_{\sum_{i=1}^{k} r_i = n} A(r_1) \times \ldots \times A(r_k) \to A(n)$$

The unit for A is given by the unit of t. Associativity and the unit law hold because the their counterparts for t. $\square$

If we can replace 1 by any category X, we obtain a correspondence between coloured plain operads or plain multicategories (cf. Section 3.2.3 of Chapter 3). We record this fact

Proposition 6.7. *There is a one-to-one correspondence between multicategories and monads in* kl$(\tilde{M})$ *on discrete* X*.*

Sketch of Proof. Let $\mathcal{B}$ be a multicategory with a class of objects X. We construct a monad $t : X \nrightarrow X$ in kl$(\tilde{M})$ from $\mathcal{B}$. First note that t needs to assign a list of objects $x_1 \otimes \ldots \otimes x_n, x$ a set of morphisms. We do this by setting t to be a functor

$$MX^{op} \times X \to \mathsf{Set}, \quad t(x_1 \otimes \ldots \otimes x_n, x) = \mathcal{B}(x_1 \otimes \ldots \otimes x_n, x).$$

Functoriality is trivial since X is discrete. We explain how to define the multiplication for t

$$t^2 = \ X \overset{t}{\nrightarrow} MX \overset{\tilde{M}t}{\nrightarrow} MMX \overset{\tilde{m}_X}{\nrightarrow} MX$$

Let $l = x_1 \otimes \ldots \otimes x_n \in MX$ and let $x \in S$

$$\begin{aligned}
t^2(l,x) &= \int^{l' \in MX} t(l',x) \times \tilde{m}_X \cdot \tilde{M}t(l,l') \\
&\simeq \int^{l'=x'_1\otimes\ldots\otimes x'_n \in MX} t(l',x) \times \int^{s_1\otimes\ldots\otimes s_n \in MMX} \tilde{M}t(s_1 \otimes \ldots \otimes s_n, l') \times \tilde{m}_X(l, s_1 \otimes \ldots \otimes s_n) \\
&\simeq \int^{l' \in MX} t(l',x) \times \int^{s_1\otimes\ldots\otimes s_n \in MMX} \tilde{M}t(s_1 \otimes \ldots \otimes s_n, l') \times \underbrace{MX(l, m_X(s_1 \otimes \ldots \otimes s_n))}_{\neq\emptyset \iff s_1\otimes\ldots\otimes s_n \text{ partitions } l} \\
&\simeq \int^{l' \in MX} t(l',x) \times \int^{s_1\otimes\ldots\otimes s_n \text{ partitions } l} t(s_1,x'_1) \times \ldots \times t(s_n,x'_n) \\
&= \coprod_{l' \in MX} t(l',x) \times \coprod_{s_1\otimes\ldots\otimes s_n \text{ partitions } l} t(s_1,x'_1) \times \ldots \times t(s_n,x'_n).
\end{aligned}$$

For every partition $s_1 \otimes \ldots \otimes s_n$ of l,every $x \in X$ and every list $l' = x'_1 \otimes \ldots \otimes x'_n$ the multicategory $\mathcal{B}$ gives a passage

$$t(l',x) \times t(s_1,x'_1) \times \ldots \times t(s_n,x'_n) \to t(l,x).$$

and hence we also obtain a passage $t^2(l,x) \to t(l,x)$.

The rest of the proof is similar to the proof of Proposition 6.6. □

Finally, we can consider the case where X is not discrete. The a monad $t : X \nrightarrow X$ in $\mathsf{kl}(\tilde{M})$ again gives a multicategory, but one equipped by right actions of X-morphisms and left actions of MX-morphisms. We analyse this scenario in the Example below.

Example 6.8. Suppose that X is a non-discrete category and let $t : X \nrightarrow X$ be a monad on X in $\mathsf{kl}(\tilde{M})$. We think of

$$t(x_1 \otimes \ldots \otimes x_n, x)$$

as a set of t-morphisms from $x_1 \otimes \ldots \otimes x_n$ to x (see Example 6.4 for more on this interpretation). Then the isomorphisms

$$\int^{x' \in X} t(x_1 \otimes \ldots \otimes x_n, x') \times X(x',x) \xrightarrow{\lambda} t(x_1 \otimes \ldots \otimes x_n, x)$$

and

$$\int^{x'_1\otimes\ldots\otimes x'_n \in MX} MX(x_1 \otimes \ldots \otimes x_n, x'_1 \otimes \ldots \otimes x'_n) \times t(x'_1 \otimes \ldots \otimes x'_n, x') \xrightarrow{\rho} t(x_1 \otimes \ldots \otimes x_n, x)$$

form left- and right-actions on the class of t-morphisms. For example, given $\varphi \in t(x_1 \otimes \ldots \otimes x_n, x')$ and $f : x' \to x$ in X,

$$\lambda(\psi, f) = t(x_1 \otimes \ldots \otimes x_n, f)(\varphi) =: f * \varphi$$

Similarly, ρ lets us define $\varphi * (f_1, \ldots, f_n)$.

Following the arguments given in Example 6.4 we can show that the multiplication m of the monad t allows to compose t morphisms via

$$(\varphi_1 \otimes \ldots \otimes \varphi_n) \Box \varphi := m(\varphi_1 \otimes \ldots \otimes \varphi_n, \varphi)$$

$\Box$, $*$ and $\cdot$ are compatible. For example

$$(f_1 \otimes \ldots \otimes f_n)\Box\varphi = (f_1 \otimes \ldots \otimes f_n) * \varphi,$$
$$(f)\Box g = (f) * g = f \cdot g.$$

While monads on non-discrete X provide the most flexible framework for defining theories, they do not correspond to any standard notion from Categorical Universal Algebra. Consequently, we shall mainly focus on the discrete case in the remainder of this work.

6.2.2 Monads in $\mathsf{kl}(\tilde{S})$

Following the plain case, we begin with a proposition.

Proposition 6.9 (Hyland)**.** *There is a one-to-one correspondence between monads* $t : 1 \nrightarrow 1$ *in* $\mathsf{kl}(\tilde{S})$ *and symmetric operads.*

Sketch of Proof. The corresponded between monads t and operads A is given in Proposition 6.6. Here we explain how the monads gives rise to the actions of the symmetric groups and to equivariance. Let t be given. We produce the action

$$\rho_k : t(k, 1) \times S_k \to t(k, 1),$$

from

$$\begin{aligned} t(k, 1) \times S_k &= t(k, 1) \times S1(k, k) \\ &\hookrightarrow \int^{k' \in S1} t(k', 1) \times S1(k, k') \overset{\cong}{\Rightarrow} t(k, 1). \end{aligned}$$

Explicitly,

$$\rho_k(\varphi, \sigma) = t(\sigma, 1)(\varphi)$$

The functoriality of $t(-, 1)$ implies the axioms of a group actions.

To understand equivariance, we need to compute t^2.

$$
\begin{aligned}
t^2(m,1) &= \tilde{m}_1 \cdot \tilde{S}t \cdot t(m,1)\\
&\simeq \int^{n\in S1} t(n,1)\times \int^{k_1\otimes\ldots\otimes k_n\in SS1} \tilde{S}t(k_1\otimes\ldots\otimes k_n,k)\times \tilde{m}_1(m,k_1\otimes\ldots\otimes k_n)\\
&\simeq \int^{n\in S1} t(n,1)\times \int^{k_1\otimes\ldots\otimes k_n\in SS1} \int^{k'_1,\ldots,k'_n\in S1} t(k'_1,1)\times\ldots\times t(k'_n,1)\times\\
&\qquad SS1(k_1\otimes\ldots\otimes k_n, k'_1\otimes\ldots\otimes k'_n)\times S1(m,m_1(k_1\otimes\ldots\otimes k_n))\\
&\simeq \int^{n\in S1} t(n,1)\times \int^{k'_1,\ldots,k'_n\in S1} \int^{k_1\otimes\ldots\otimes k_n\in SS1} t(k'_1,1)\times\ldots\times t(k'_n,1)\times\\
&\qquad SS1(k_1\otimes\ldots\otimes k_n, k'_1\otimes\ldots\otimes k'_n)\times S1(m,m_1(k_1\otimes\ldots\otimes k_n))\\
&\simeq \int^{n\in S1} t(n,1)\times \int^{k'_1,\ldots,k'_n\in S1} t(k'_1,1)\times\ldots\times t(k'_n,1)\times S1(m,\sum_{i=1}^{n} k_i)\\
&\simeq \int^{n\in S1} t(n,1)\times \int^{k'_1+\ldots+k'_n=m} t(k'_1,1)\times\ldots\times t(k'_n,1)\times S1(m,m)
\end{aligned}
$$

Let $x\in t(n,1)$, $\big((x_1,\ldots,x_n),s\big)\in \tilde{m}_1\cdot\tilde{S}t(m,n)$ and $\sigma\in S1(n,n)$. We claim that equivariance follows from the fact that in $t^2(m,1)$ the following are equal

$$
\big(t(\sigma,1)(x),((x_1,\ldots,x_n),s)\big) = \big(x,\tilde{m}_1\cdot\tilde{S}t(m,\sigma)((x_1,\ldots,x_n),s)\big) \tag{6.3}
$$

Indeed it is easy to see that the left-hand-side of Equation 6.3 is corresponds to the lower-left passage of diagram 3.4. And since action $\tilde{m}_1\cdot\tilde{S}t(m,\sigma)((x_1,\ldots,x_n),s)$ corresponds to first permuting the operations x_i and then precomposing them with the permutation σ, the RHS corresponds to the upper-right passage of that diagram. □

As in the previous section we have

Proposition 6.10. *The is a one-to-one correspondence between symmetric multicategories and monads in* kl$(\tilde{S})$ *on discrete* X.

The proof is analogous to the proof of Proposition 6.7.

When X is not discrete, something interesting happens. The class of morphisms introduced by t interacts nicely not only with the symmetries of SX but also with the non-identity morphisms of X. The two interactions are very different. The interaction with symmetries is a derivative of the setting kl$(\tilde{S})$ and is entirely independent of X, while the interaction with the morphisms of X follows the pattern described in Example 6.8 and depends only on X itself.

6.2.3 Monads in kl$(\tilde{F})$

In this section we are finally able to precisely define *Cartesian Operads* (cf. Example 3.9). We shall show that all the properties we sketched in Example 3.9 can be derived from the this formal definition.

Definition 6.11. *A* Cartesian operad *is a monad* $t : 1 \nrightarrow 1$ *in* kl$(\tilde{F})$.

Let t be a Cartesian operad. Then, since $M1 \hookrightarrow F1$, t is an operad. As a profunctor t is given by

$$t : \mathbb{F}^{\mathrm{op}} \times 1 \to \mathsf{Set}$$

We interpret $t(n, 1)$ as the set of operations of arity n. Right actions of finite-product structure on the operations of t are given by the isomorphisms

$$t(k,1) \times F1(n,k) \hookrightarrow \int^{k \in F1} t(k,1) \times F1(n,k) \simeq t(n,1)$$

which explicitly send the pair

$$(x \in t(k,1), \varphi : n \to k) \mapsto t(\varphi, 1)(x) \in t(n,1)$$

Action axioms follow immediately from the functoriality of $t(-, 1)$.

We can compute

$$\begin{aligned} t^2(m,1) &= \int^{n \in F1} t(n,1) \times \int^{k_1,\ldots,k_n \in F1} t(k_1,1) \times \ldots \times t(k_r,1) \times F1(m, \sum_{i=0}^{n} k_i) \\ &= \int^{n \in F1} t(n,1) \times (\tilde{m}_1 \cdot \tilde{F}t)(m,n) \end{aligned}$$

Note that the Cartesian analogue of equivariance follows from the fact that for operations $x \in t(n', 1)$, $x_i \in t(k_1, 1)$, $i = 1, \ldots, n$ and morphisms $\psi \in F1(m, \sum_{i=0}^{n} k_i)$ and $\varphi \in F1(n, n')$,

$$\big(t(\varphi,1)(x), (\psi, x_1, x_1, \ldots, x_n)\big) = \big((x, \tilde{m}_1 \cdot \tilde{F}t(m,\varphi)(\psi, x_1, \ldots, x_n)\big) \tag{6.4}$$

To illustrate this point, we examine the situation when $\varphi = \kappa : 1 \to 1 \times 1$ is the copying operation. Assume, for the sake of clarity, that $m = 3 = 1 \times 1 \times 1$ and $\psi = \mathsf{id}_3$. Let $x_0 \in t(3,1) = \tilde{F}t([3],1)$[1]; we wish to understand the expression

$$\tilde{F}t(3,\kappa)(x_0) := \tilde{F}t(3,\kappa)(\mathsf{id}_3, [x_0]) \in \tilde{F}t([3], 2)$$

Recall from Chapter 5 that

$$\tilde{F}t = \; F1 \xrightarrow{Ft} FPF1 \xrightarrow{\xi_{F1}} PF^2 1 \, .$$

and that this composite works as follows

$$\begin{aligned} 1 \xrightarrow{\kappa} 1 \times 1 \; &\mapsto \; t(-,1) \xrightarrow{Ft(\kappa)} t(-,1) \times t(-,1) \\ &\mapsto \; \textstyle\int^{n \in F1} t(n,1) \times F^1(-,n) \xrightarrow{\tilde{F}t(\kappa)} \textstyle\int^{n_1, n_2 \in F1} t(n_1,1) \times t(n_2,1) \times F^2 1(-, n1 \times n2) \end{aligned}$$

where the natural transformation $\tilde{F}t(\kappa)$ is given by

$$\tilde{F}t(\kappa)_{k_1 \times \ldots \times k_p}\big((x, \pi : k_1 \times \ldots \times k_n \to n)\big) = \big((x,x), (k_1 \times \ldots \times k_p \xrightarrow{\pi} n \xrightarrow{\kappa_n} n \times n)\big)$$

(here κ_n is derived from κ as described in Example 3.9 of Chapter 3.)

[1]We write $[m]$ for the singleton list containing m

Thus $\tilde{F}t(3,\kappa)(x_0) = \big((x_0, x_0), \kappa_3\big)$ and this passage is visualised in Diagram 6.1.

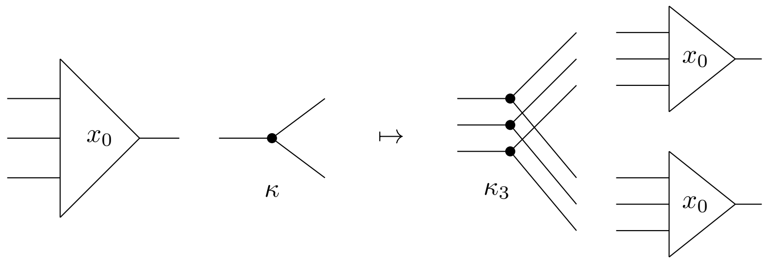


Diagram 6.1: The actions of $\tilde{F}t(3,\kappa)$

With $\varphi = \kappa$, we can now visualise the equality in Equation 6.4 in Diagram 6.2. Note that this diagram is

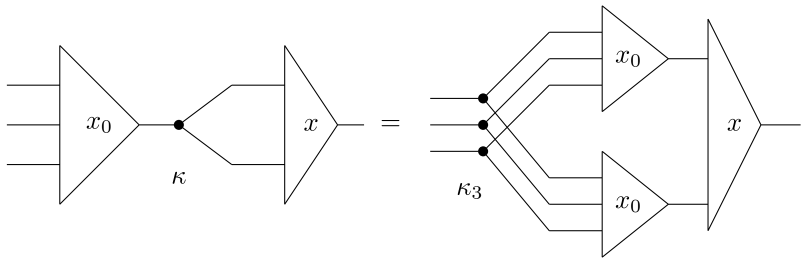


Diagram 6.2: The equality of equation 6.4

identical to Diagram 3.8 of Chapter 3.

Similar equations can be derived for more general $\varphi \in F1$.

Thus equivariance - which is the most complicated component in the definition of a Cartesian operad - can be derived from our setting. Indeed, its complex combinatorial nature is elegantly captured by $\tilde{F}t$.

6.3 Concrete theories in $\mathsf{kl}(\tilde{T})$

We saw that when the 2-monad T distributes over the presheaf monad, theories of type T can can be expressed as monads in $\mathsf{kl}(\tilde{T})$. In this section we look at concrete example of such theories.

Example 6.12. We give the theory for commutative semi-groups in symmetric monoidal categories. Thus our aim is to define a monad

$$t : 1 \nrightarrow 1 \in \mathsf{kl}(\tilde{S}).$$

This is equivalent to giving a functor $t : S1^{op} \times 1 \to \mathsf{Set}$, which we define by

$$t(n,1) = \{n\} \text{ if } n > 0, \text{ and } t(0,1) = \emptyset.$$

We interpret the single element in $t(n,1)$ as the single operation of arity n. So, for example, the element

$$2 \in t(2,1)$$

is the multiplication of our theory.

Next we need to define the action of t on symmetries $\sigma : n \to n \in S1$. Since, $t(n,1)$ is a singleton, we are forced to set $t(\sigma, 1)$ to identity. This makes our semi-group commutative.

There is little choice for the multiplication

$$m : t^2 \to t$$

$m_(n,1)$ sends each element of $t^2(n,1)$ to the single element of $t(1)$.

Example 6.13. A *pointed set* is a set X together with a chosen element $x_0 \in X$. We can represent the theory of pointed sets as a monad on

$$s : 1 \nrightarrow 1$$

in $\mathsf{kl}(\tilde{S})$.

As a functor, $s : S1^{\mathrm{op}} \times 1 \to \mathsf{Set}$ sends each $n \geq 2$ to the empty set and,

$$s(0,1) = \{0\} \text{ and } s(1,1) = \{1\}.$$

The definition of s on symmetries is self-evident. To define the multiplication, we note that

$$s^2(n,1) = \emptyset, \text{ if } n \geq 2,$$

and if case $n = 0$ or $n = 1$, $s^2(n,1) \simeq s(n,1)$, which gives the multiplication.

Example 6.14. Let X be a symmetric monoidal category and let $x_0 \in X$ be an object of X. We define the *theory of morphisms on* x_0 to be an arrow

$$x_0 : 1 \nrightarrow 1$$

in $\mathsf{kl}(\tilde{S})$ given as a functor by $x_0(n,1) = X(x_0^{\otimes n}, x_0)$. The multiplications and unit are given by the compositions and identity in X. This theory corresponds to the endomorphisms operad of Example 3.7.

The above example allows us to define models of theories. Recall from Chapter 3 that a model on an object $x \in X$ of an operad A is a morphism of operads $A \to \mathsf{End}(x)$. The same idea can be used to define models of monads $t : 1 \nrightarrow 1$ in $\mathsf{kl}(\tilde{T})$. Given a T-category X and an object $x_0 \in X$, a *model of t on x* is a monad morphism

$$\alpha : t \to x_0.$$

where x_0 is the monad of Example 6.14.

Example 6.15. Let t be the theory of Example 6.12. Let X be a symmetric monoidal category and let x_0 be any object in X. Write x_0 for the monad on 1 induced by x_0. Then a monad morphism

$$\alpha : t \to x_0$$

gives, for every $n \in S1$,

$$\alpha_n : t(n,1) \to x_0(n,1) = X(x_0^{\otimes n}, x_0)$$

the concrete implementation in X of all operations of arity n. The fact that α is a monad morphism, means that all the axioms of the theory t hold also hold for these concrete implementations in X.

6.4 Distributive Laws in 2-categories

One advantage of the internal monadic view of operads and categories is that it comes with a tool for combining these structures. We already noted that given any two monads, a distributive law specifies a method for creating the composite of these monads. Hence (internal) distributive laws give us a single mechanism for combining categories, PROPs, PROs, Lawvere theories and operads of various types.

Let $s = (s, \mu_s, \eta_s)$ and $t = (t, \mu_t, \eta_t)$ be monads in a bicategory $\mathcal{K}$. An *internal distributive law* of s over t is a 2-cell

$$\lambda : s \cdot t \to t \cdot s$$

such that

$$\begin{tikzcd} & t \arrow[dl, "t\eta_s"'] \arrow[dr, "\eta_s t"] & \\ ts \arrow[rr, "\lambda"] & & st \\ & s \arrow[ul, "\eta_t s"] \arrow[ur, "s\eta_t"'] & \end{tikzcd} \qquad \begin{tikzcd} tss \arrow[r, "\lambda s"] \arrow[d, "t\mu_s"'] & sts \arrow[r, "s\lambda"] & sst \arrow[d, "\mu_s t"] \\ ts \arrow[rr] & & st \\ tts \arrow[u, "\mu_t s"] \arrow[r, "t\lambda"'] & tst \arrow[r, "\lambda t"'] & stt \arrow[u, "s\mu_t"'] \end{tikzcd}$$

Example 6.16. Let $\mathcal{C}$ and $\mathcal{C}'$ be categories with $\mathrm{ob}\mathcal{C} = X = \mathrm{ob}\mathcal{C}'$. Then $\mathcal{C}$ and $\mathcal{C}'$ are monads on X in Span - we again write $\mathcal{C}$ and $\mathcal{C}'$ for those monads. We write $\mathcal{C} \otimes \mathcal{C}'$ for the span $\mathcal{C} \times_X \mathcal{C}'$. A distributive law between $\mathcal{C}$ and $\mathcal{C}'$ is a 2-cell

$$l : \mathcal{C} \otimes \mathcal{C}' \to \mathcal{C}' \otimes \mathcal{C}.$$

This distributive law l sends pairs of compossible morphisms to pairs of compossible morphisms. Given a pair of compossible morphisms $(f, g) \in \mathcal{C} \otimes \mathcal{C}'$ we write

$$l(f, g) = (g_{\mathcal{C}'} f, g_{\mathcal{C}} f)$$

and denote the domain of $g_{\mathcal{C}'} f$ and the codomain of $g_{\mathcal{C}} f$ by $g_l f$. This allows to capture l diagrammatically

$$\begin{tikzcd} A \arrow[r, "f \in \mathcal{C}'"] \arrow[d, "g_{\mathcal{C}} f"'] & B \arrow[d, "g \in \mathcal{C}"] \arrow[dl, "l"'] \\ g_l f \arrow[r, "g_{\mathcal{C}'} f"'] & C \end{tikzcd}$$

(Note that the above is not a commutative diagram, but only a depiction of l).

The defining conditions for a distributive law translate into preservation of the unit and composition of $\mathcal{C}$ and $\mathcal{C}'$. The above notations allows us to capture this precisely as,

$$f_{\mathcal{C}} 1_A = f, \; f_{\mathcal{C}'} 1_A = 1_B, \; f_l 1_A = B.,$$

$$(1_B)_{\mathcal{C}} f = 1_A,\ (1_B)_{\mathcal{C}'} f = f,\ (1_B)_l f = A,$$

for the interaction with the units and

$$\begin{aligned} h_{\mathcal{C}'}(k \cdot f) &= h_{\mathcal{C}'} k \cdot (h_{\mathcal{C}} k)_{\mathcal{C}'} f \\ h_{\mathcal{C}}(k \cdot f) &= (h_{\mathcal{C}} k)_{\mathcal{C}} f \\ (h_{\mathcal{C}} k)_l f &= h_l(k \cdot f) \end{aligned} \tag{6.5}$$

and

$$\begin{aligned} (h \cdot g)_{\mathcal{C}'} f &= h_{\mathcal{C}'}(g_{\mathcal{C}'} f), \\ (h \cdot g)_{\mathcal{C}} f &= h_{\mathcal{C}}(g_{\mathcal{C}'} f) \cdot g_{\mathcal{C}} f, \\ (h \cdot g)_l f &= h_l(g_{\mathcal{C}'} f), \end{aligned} \tag{6.6}$$

for the interaction with composition. See [Akhvlediani, 2009] for full details.

Example 6.17. One can similarly define a distributive law between PROs. We saw in Example **??** that PROs are monads in $\mathsf{Span}(\mathsf{Mon})$. Hence a distributive law between PROs is a distributive law between the associated monads in this 2-category. Its easy to spell this out in detail: such distributive law amount to giving a distributive law between the underlying categories, such that the function l of Example **??** preserves the tensor product. More precisely, given PROs $\mathbb{S}$ and $\mathbb{T}$, a distributive law between then is a 2-cell

$$L : \mathbb{S} \to \mathbb{T}$$

in $\mathsf{Span}(\mathsf{Mon})$, such that l is distributive law of the underlying categories $\mathbb{S}$ and $\mathbb{T}$ and for any pairs of composible morphisms $(g, f), (g', f') \in \mathbb{T} \otimes \mathbb{S}$

$$\begin{aligned} g \otimes g'_{\mathbb{S}} f \otimes f' &= g_{\mathbb{S}} f \otimes g'_{\mathbb{S}} f' \\ g \otimes g'_{\mathbb{T}} f \otimes f' &= g_{\mathbb{T}} f \otimes g'_{\mathbb{T}} f' \\ g \otimes g'_l f \otimes f' &= g_l f \otimes g'_l f' \end{aligned}$$

Example 6.18. PROPs are monads in $\mathsf{Prof}(\mathsf{Mon})$ (cf. Example 6.5). A distributive law between PROPs is a distributive law between the underlying PROs such that whenever there exists $\pi \in \mathbb{P}$ making the upper-right triangles in the following diagram commute, there exist a $\pi' \in \mathbb{P}$ making the lower-left triangles commute

$$\begin{tikzcd} & & & \cdot \arrow[dl, bend left, "\tau'"] \\ & \cdot \arrow[r, "\sigma"] \arrow[d, "\tau_{\mathbb{T}}\sigma"'] \arrow[urr, bend left, "\sigma'"] \arrow[ddl, bend right, "\tau'_{\mathbb{T}}\sigma'"'] & \cdot \arrow[d, "\tau"] \arrow[ur, "\pi"] \arrow[dl, "l"'] & \\ & \cdot \arrow[r, "\tau_{\mathbb{S}}\sigma"'] \arrow[dl, "\pi'"] & \cdot \arrow[dll, bend left, "\tau'_{\mathbb{S}}\sigma'"] & \\ \cdot & & & \end{tikzcd}$$

See Theorem 6 of [Akhvlediani, 2009] for a proof.

Example 6.19. In [Akhvlediani, 2010], the author showed that Lawvere theories can be viewed as monads over $\mathbb{F}^{\mathrm{op}}$ in $\mathsf{Prof}(\mathsf{Mon})$. As such, those Lawvere theories can be composed to produce a new monad on $\mathbb{F}^{\mathrm{op}}$. It turns out that this new monad is also a Lawvere theory. Consult the slides in the reference for a complete proof.

Example 6.20. In Examples 6.12 and 6.13 we defined the theory for commutative semi-groups, t and the theory for pointed sets s. We now show that those theories can be combined with a distributive law to produce the theory of commutative monads.

The distributive law is a 2-cell in $\mathsf{kl}(\tilde{S})$

$$\lambda : t \cdot s \to s \cdot t.$$

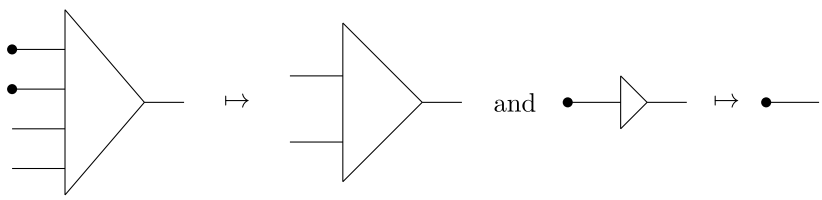
To understand λ, we first compute its domain and codomain: the domain is

$$t \cdot s(n,1) = \int^{k \in S1} t(k,1) \times \int^{k'_1 + \ldots + k'_r = n} s(k'_1) \times \ldots \times s(k'_r) \times S1(n,n);$$

and codomain is given by

$$\begin{aligned} s \cdot t(n,1) &= \int^{k \in \mathbb{P}} s(k,1) \times \int^{k'_1 + \ldots + k'_k = n} t(k'_1) \times \ldots \times t(k'_k) \times S1(n,n) \\ &\simeq s(0) \coprod s(1) \times t(n) \times S1(n,n). \end{aligned}$$

The action of λ_n can now be seen as



The distributive law λ turns the composite 1-cell $s \cdot t$ into a monad with multiplication given by

$$(s \cdot t) \cdot (s \cdot t) \to (t \cdot t) \cdot (s \cdot s) \to s \cdot t$$

Its operations are given by

$$s \cdot t(n,1) = \{n\}, \text{ for all } n \in S1.$$

The distributive law λ transforms the 0-arity operation $0 \in s \cdot t(0)$ into the unit for the multiplication $2 \in s \cdot t(2)$, making the composite monad the theory for commutative monoids, as desired.

6.5 Preservation of Internal Structure

We briefly note that internal structures in 2-categories are preserved by 2-functors. We shall use these facts in Chapter **??** to construct theories of one type from theories of another type.

Proposition 6.21. *Let $T : \mathcal{K} \to \mathcal{M}$ be a 2-functor between 2-categories. Then T preserves internal monads.*

Proposition 6.22. *A 2-functor $T : \mathcal{K} \to \mathcal{M}$ preserves internal distributive laws.*

Both these results are well known and their proofs are straightforward.

Chapter 7

Extensions to $\mathsf{kl}(P)$

In the previous chapter we saw that the tools of Categorical Universal Algebra can be viewed as monads in various bicategories. In particular, operads of various types are monads in Kleisli bicategories of extensions of Cat-monads. We also saw that the notions of *type* is captured by a Cat monads that distribute over the presheaf monad.

The aim of the next three chapters is the understand how relationships between types lead to relationships between theories of these types. To do this, we need to understand now only how monads distribute over the presheaf monad P, but how 2-natural transformation and modifications distribute. We do this in this chapter.

More precisely, in this chapter we construct two monoidal bicategories, DL^P and Ext_P. The former consists of 2-functors, 2-natural transformations and modifications that distributive over P. The latter consists of extensions of 2-functors, 2-natural transformations and modifications to $\mathsf{kl}(P)$.

We show that both DL^P and Ext_P are monoidal bicategories. Furthermore, we show that there is a monoidal biequivalence between them. While this result is easy to state, its proof is fairly involved. The proof was broken down into manageable components. First we explain how the 0-, 1- and 2-cells of DL^P map to the corresponding structures in Ext_P. We prove in full detail that our constructions indeed end-up in Ext_P. Certain coherence conditions are verified with particular care. We also explain how Ext_P maps back into DL^P, albeit with less detail. We prove 2-functoriality of both passages and show that they preserve the monoidal structure.

The chapter ends with two important corollaries. The first one states that if S is a *monad* that distributes over P, then its extension to $\mathsf{kl}(P)$ is also a monad. The second one shows that when α is a monad morphism that distributes over P, then its extension is also a monad morphism. We shall use these two corollaries in the next chapter to construct a 2-functor between $\mathsf{kl}(\tilde{S})$ and $\mathsf{kl}(\tilde{T})$ that will send theories of type S to theories of type T.

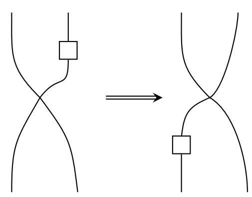
The monoidal biequivalence we construct should we thought of as a categorification of Beck's Theorem [Beck, 1969]. This categorification was started in [Tanaka, 2005]. There Tanaka showed that there is a very similar biequivalence between a 2-category dual to DL^P and a 2-category of lifts of 2-functors to the category of algebras of the monad P. Thus Tanaka's thesis categorified one half of Beck's Theorem. In this chapter we categorify the other half.

7.1 The monoidal 2-category of distributive laws

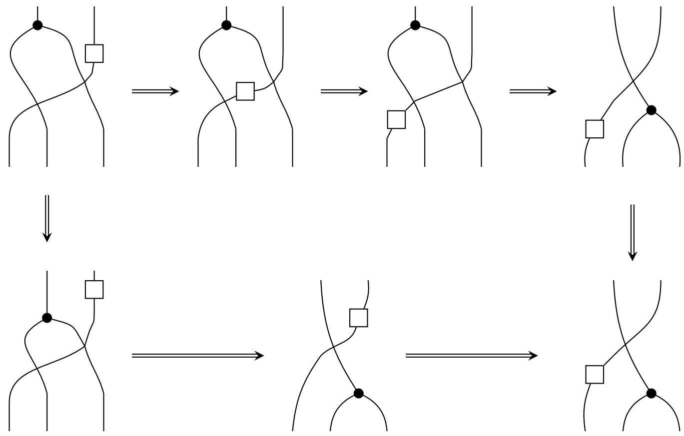
Let $\mathcal{K}$ be a 2-category and P be a 2-monad on $\mathcal{K}$. We define the 2-category DL^P of distributive laws over P as follows

- 0-cells are pairs (T, λ) with $T : \mathcal{K} \to \mathcal{K}$ an 2-functor and $\lambda : TP \to PT$ a 2-distributive law.

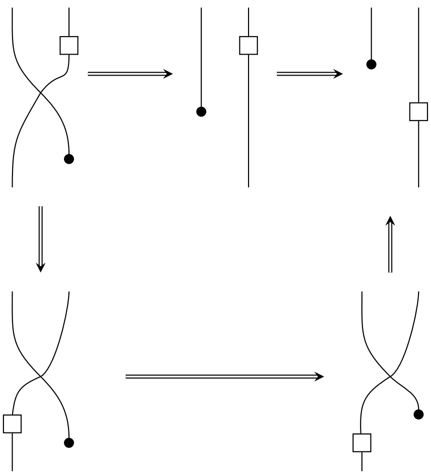
- 1-cells are pairs
$$(\alpha, \alpha^*) : (T, \lambda) \to (S, \rho)$$
where $\alpha : T \to S$ is a 2-natural transformation and α^* is a modification
$$\begin{tikzcd} SP \arrow[r, "\rho"] \arrow[d, "\alpha P"'] \arrow[dr, phantom, "\Downarrow\alpha^*"] & PS \arrow[d, "P\alpha"] \\ TP \arrow[r, "\lambda"'] & PT \end{tikzcd}$$
satisfying the following coherence conditions:
$$\begin{tikzcd} SP^2 \arrow[rr, "\rho P"] \arrow[dr, "\alpha P^2"] \arrow[dd, "S\mu"'] & & PSP \arrow[rr, "P\rho"] \arrow[dr, "P\alpha P"] \arrow[l, phantom, "\Downarrow\alpha^* P"] & & P^2S \arrow[dr, "P^2\alpha"] \arrow[l, phantom, "\Downarrow P\alpha^*"] & \\ & TP^2 \arrow[rr, "\lambda P"] \arrow[dd, "T\mu"] & & PTP \arrow[rr, "P\lambda"] \arrow[d, phantom, "\Downarrow"] & & P^2T \arrow[dd, "\mu T"] \\ SP \arrow[dr, "\alpha P"'] \arrow[ur, phantom, "\Downarrow"] & & & & & \\ & TP \arrow[rrrr, "\lambda"'] & & & & PT \end{tikzcd} = \begin{tikzcd} SP^2 \arrow[r, "\rho P"] \arrow[dd, "S\mu"'] & PSP \arrow[r, "P\rho"] \arrow[d, phantom, "\Downarrow"] & P^2S \arrow[dr, "P^2\alpha"] \arrow[dd, "\mu S"] & \\ & & & P^2T \arrow[dd, "\mu T"] \arrow[dl, phantom, "\Downarrow"] \\ SP \arrow[rr, "\rho"] \arrow[dr, "\alpha P"'] & & PS \arrow[dr, "P\alpha"] \arrow[dl, phantom, "\Downarrow\alpha^*"] & \\ & TP \arrow[rr, "\lambda"'] & & PT \end{tikzcd} \tag{7.1}$$
and
$$\begin{tikzcd} & SP \arrow[dr, "\rho"] \arrow[d, phantom, "\Downarrow"] & \\ S \arrow[ur, "S\eta"] \arrow[rr, "\eta S"] \arrow[dd, "\alpha"'] & & PS \arrow[dd, "P\alpha"] \\ & \Downarrow & \\ T \arrow[rr, "\eta T"'] & & PT \end{tikzcd} \qquad \begin{tikzcd} & SP \arrow[dr, "\rho"] \arrow[dd, "\alpha P"] & \\ S \arrow[ur, "S\eta"] \arrow[dd, "\alpha"'] & \Downarrow \quad \Downarrow\alpha^* & PS \arrow[dd, "P\alpha"] \\ & TP \arrow[dr, "\lambda"] \arrow[d, phantom, "\Downarrow"] & \\ T \arrow[ur, "T\eta"] \arrow[rr, "\eta T"'] & & PT \end{tikzcd}$$
We visualise α^* as a rewrite as follows



These coherence conditions have an intuitive graphical interpretation: the first one translates to



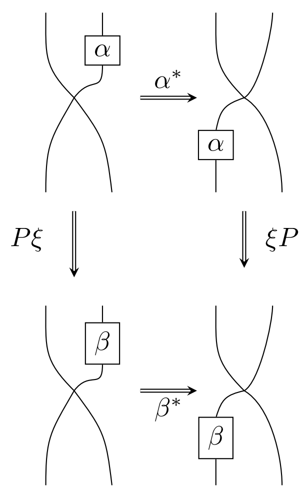
and the second translates to



- 2-cells are modifications $\xi : \alpha \to \beta$ such that

$$\begin{tikzcd} SP \arrow[r, "\rho"] \arrow[d, bend right, "\alpha P"'] \arrow[d, bend left, "\beta P"] \arrow[d, phantom, "\Downarrow"] \arrow[dr, phantom, "\Downarrow"] & PS \arrow[d, "P\beta"] \\ TP \arrow[r, "\lambda"'] & PT \end{tikzcd} \quad = \quad \begin{tikzcd} SP \arrow[r, "\rho"] \arrow[d, "\alpha P"'] \arrow[dr, phantom, "\Downarrow P\alpha"] & PS \arrow[d, bend right, "P\alpha"'] \arrow[d, bend left, "P\beta"] \arrow[d, phantom, "\Downarrow"] \\ TP \arrow[r, "\lambda"'] & PT \end{tikzcd} \tag{7.2}$$

This condition can be seen as requiring that any rewrite in DL^P is compatible with rewrites of the form α^*. Visually:



Proposition 7.1. *The data in the definition of* DL^P *forms a 2-category.*

Proof. We define vertical and horizontal composition. For 1-cells

$$(S,\rho) \xrightarrow{(\alpha,\alpha^*)} (T,\lambda) \xrightarrow{(\beta,\beta^*)} (W,\delta)$$

we set $(\beta,\beta^*) * (\alpha,\alpha^*) := (\beta\cdot\alpha, (\beta\cdot\alpha)^*)$ where $(\beta\cdot\alpha)^*$ is given by

$$\begin{tikzcd}
SP \arrow[r, "\rho"] \arrow[d, "\alpha P"'] \arrow[dr, phantom, "\Downarrow"] & PS \arrow[d, "P\alpha"] \\
TP \arrow[r, "\lambda"] \arrow[d, "\beta P"'] \arrow[dr, phantom, "\Downarrow"] & PT \arrow[d, "P\beta"] \\
WP \arrow[r, "\delta"] & PW
\end{tikzcd}$$

Horizontal and vertical composition of 2-cells is given by standard composition of modifications. ☐

Furthermore, DL^P has the structure of a monoidal 2-category. We describe the structure of the tensor on 0-, 1- and 2-cells.

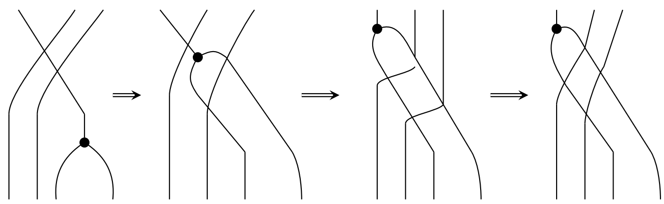
- Given two 0-cells (S,ρ) and (T,λ), we define

  $$(S,\rho)\otimes(T,\lambda) := (S\cdot T, \rho\otimes\lambda)$$

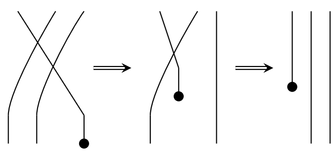
  where the distributive law $\rho\otimes\lambda : STP \to PTS$ is given by the composite

  $$\rho\otimes\lambda := \;STP \xrightarrow{S\lambda} SPT \xrightarrow{\rho T} PTS$$

  the modifications for $\rho\otimes\lambda$ are given by

  and

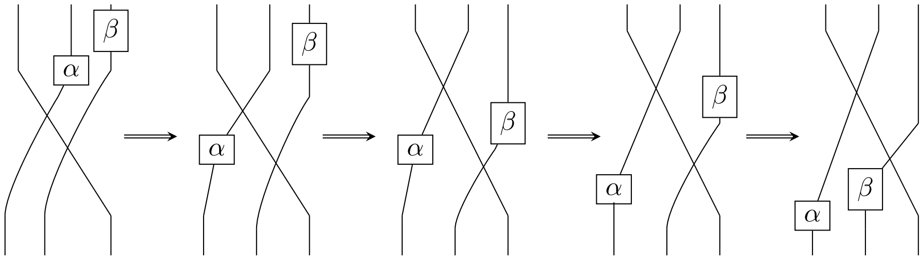
  

  The fact the $\rho\otimes\lambda$ is a 2-distributive law follows from the dual of Proposition 6.9 [Tanaka, 2005]

- Given 1-cells $(\alpha,\alpha^*) : (S,\rho) \to (T,\lambda)$ and $(\beta,\beta^*) : (S',\rho') \to (T,'\lambda')$ in DL^P, we define $\alpha\otimes\beta$ to be the horizontal composite of these 2-natural transformations. For definiteness, we set

$$\alpha \otimes \beta := \; SS' \xrightarrow{\alpha S'} TS' \xrightarrow{T\beta} T'S'$$

and set $(\alpha \otimes \beta)^*$ to



- The tensor product of 2-cells in DL^P is given by the standard tensor product of modifications.

It turns out that the above definition gives rise to a 2-functor

$$\otimes : \mathsf{DL}^P \times \mathsf{DL}^P \to \mathsf{DL}^P.$$

As a consequence we have
Theorem 7.2. *The 2-category* DL^P *is a monoidal 2-category.*

7.2 The monoidal 2-category of extensions

In this section we make the idea of an *extension* precise. We begin by fixing a 2-functor P on a 2-category $\mathcal{K}$. The 2-category Ext_P of all functors that can be extended from $\mathcal{K}$ to $\mathsf{kl}(P)$ consist of the following ingredients

- 0-cells are are pairs of 2-functors $(S, \tilde{S})$ with $S : \mathcal{K} \to \mathcal{K}$ and $\tilde{S} : \mathsf{kl}(P) \to \mathsf{kl}(P)$ such that the following square commutes

$$\begin{tikzcd} \mathsf{kl}(P) \arrow[r, "\tilde{S}"] & \mathsf{kl}(P) \\ \mathcal{K} \arrow[u, "(-)_*"] \arrow[r, "S"'] & \mathcal{K} \arrow[u, "(-)_*"'] \end{tikzcd}$$

- 1-cells are pairs of 2-natural transformations

$$(\alpha, \tilde{\alpha}) : (S, \tilde{S}) \to (T, \tilde{T})$$

where $\alpha : S \to T$ and $\tilde{\alpha} : \tilde{S} \to \tilde{T}$ such that

$$\begin{tikzcd} \mathcal{K} \arrow[r, bend left, "S"] \arrow[r, bend right, "T"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathcal{K} \arrow[r, "(-)_*"] & \mathsf{kl}(P) \end{tikzcd} = \begin{tikzcd} \mathcal{K} \arrow[r, "(-)_*"] & \mathsf{kl}(P) \arrow[r, bend left, "\tilde{S}"] \arrow[r, bend right, "\tilde{T}"'] \arrow[r, phantom, "\Downarrow\tilde{\alpha}"] & \mathsf{kl}(P) \end{tikzcd}$$

That is, for every X

$$\tilde{\alpha}_X = (\alpha_X)_*.$$

- 2-cells are pairs of modifications $(\xi, \tilde{\xi}) : (\alpha, \tilde{\alpha}) \to (\beta, \tilde{\beta})$ with $\xi : \alpha \to \beta$ and $\tilde{\xi} : \tilde{\alpha} \to \tilde{\beta}$ satisfying

$$\begin{tikzcd} (-)_* \cdot S \arrow[r, bend left, "(-)_*\cdot\alpha"] \arrow[r, bend right, "(-)_*\cdot\beta"'] \arrow[r, phantom, "\Downarrow (-)_*\cdot\xi"] & (-)_* \cdot T \end{tikzcd} = \begin{tikzcd} \tilde{S} \cdot (-)_* \arrow[r, bend left, "\tilde{\alpha}\cdot(-)_*"] \arrow[r, bend right, "\tilde{\beta}\cdot(-)_*"'] \arrow[r, phantom, "\Downarrow \tilde{\xi}\cdot(-)_*"] & \tilde{T} \cdot (-)_* \end{tikzcd}$$

It is easy to check that Ext_P is a 2-category. It turns out that it is in fact a monoidal 2-category. We describe the tensor product on 0-, 1- an 2-cells:

- Given two 0-cells $(S, \tilde{S})$ and $(T, \tilde{T})$ we define

$$(S, \tilde{S}) \otimes (T, \tilde{T}) = (S \cdot T, \tilde{S} \cdot \tilde{T}).$$

  In other words, $\widetilde{S \cdot T} = \tilde{S} \cdot \tilde{T}$.

- On 1-cells, we have

$$(\alpha, \tilde{\alpha}) \otimes (\beta, \tilde{\beta}) = (\alpha \otimes \beta, \tilde{\alpha} \otimes \tilde{\beta})$$

  with the $\otimes$ on the right coming from $\mathsf{End}(\mathcal{K})$ and $\mathsf{End}(\mathsf{kl}(P))$.

- The tensor on 2-cells is also inherited from $\mathsf{End}(\mathcal{K})$ and $\mathsf{End}(\mathsf{kl}(P))$.

From arguments dual to the ones give in Proposition 6.7 in [Tanaka, 2005], we have

Theorem 7.3. *The 2-category* Ext_P *is a monoidal 2-category.*

7.3 From 2-distributive laws to extensions

7.3.1 0-cells

It is well known that an 2-distributive law $\rho : SP \to PS$ gives rise to an extension of S to $\tilde{S}$. Here we only sketch how this process works. For a complete proof see [Cheng et al., 2004].

Theorem 7.4. *Let* (S, ρ) *be a 0-cell in* DL^P*. Define* $\tilde{S} : \mathsf{kl}(P) \to \mathsf{kl}(P)$ *by*

- $\tilde{S} = S$*, on 0-cells;*

-
$$\tilde{S}\Big(\begin{tikzcd} X \arrow[r, bend left, "\varphi"] \arrow[r, bend right, "\psi"'] \arrow[r, phantom, "\Downarrow\xi"] & PY \end{tikzcd} \Big) = \begin{tikzcd} SX \arrow[r, bend left, "S\varphi"] \arrow[r, bend right, "S\psi"'] \arrow[r, phantom, "\Downarrow S\xi"] & SPY \arrow[r, "\rho_Y"] & PSY, \end{tikzcd}$$

  on 1- and 2-cells;

Then $(S, \tilde{S})$ *is a 0-cell in* Ext^P*.*

Proof. The preservation of the unit $u_X : X \to PX = X \nrightarrow X$ of $\mathsf{kl}(P)$ by $\tilde{S}$ follows from the isomorphism

$$\rho_X \cdot u_X \simeq u_{SX}.$$

Next, let $\varphi : X \nrightarrow Y$ and $\psi : Y \nrightarrow Z$ be composible 1-cells in $\mathsf{kl}(P)$. Consider the following diagram

$$\begin{tikzcd}
SX \arrow[r, "S\varphi"] & SPY \arrow[r, "\rho_Y"] \arrow[dr, "SP\psi"'] & PSY \arrow[r, "P\psi"] \arrow[d, phantom, "\Downarrow"] & PSPZ \arrow[r, "P\rho_Z"] & P^2SZ \arrow[r, "m_{SZ}"] \arrow[d, phantom, "\Downarrow"] & PSZ \\
& & SP^2Z \arrow[ur, "\rho_{PZ}"'] \arrow[rr, "Sm_Z"'] & & SPZ \arrow[ur, "\rho_Z"'] &
\end{tikzcd}$$

The top row is the composite $\tilde{S}(\psi)\cdot\tilde{S}(\varphi)$ and the bottom row is $\tilde{S}(\psi\cdot\varphi)$ and the diagram exhibits the desired isomorphism. $\square$

7.3.2 1-cells

Theorem 7.5. *Let $(\alpha, \alpha^*) : (S, \rho) \to (T, \lambda)$ be a 1-cells in DL_P. Define*

$$\tilde{\alpha}_X = \; SX \xrightarrow{\alpha_X} TX \xrightarrow{u_{TX}} PTX \; .$$

Then $(\alpha, \tilde{\alpha})$ is a 1-cell in Ext^P.

Proof. The proof can be divided into three parts

1. Definition of $\tilde{\alpha}$ and its pseudo-naturality.
2. Show that the pseudo-naturaity is natural.
3. Show that the necessary coherence conditions hold.

(1) For every 0-cell X in $\mathcal{K}$ we define a 1-cell

$$\beta_X := \; SX \xrightarrow{\alpha_X} TX \xrightarrow{u_{TX}} PTX.$$

The β_X's will form our pseudo-natural transformation $\tilde{\alpha}$.

We proceed to define the pseudo-naturality for β. At each 1-cell $\varphi : X \to Y$ in $\mathsf{kl}(P)$ the pseudo-naturality 2-cell

$$\beta_{\varphi}^{X,Y} : \tilde{T}\varphi \cdot \beta_X \to \beta_Y \cdot \tilde{S}\varphi$$

is given by the following pasting diagram

$$\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "u_{SX}"'] \arrow[ddr, bend right, "S\varphi"'] & TX \arrow[r, "u_{TX}"] \arrow[d, phantom, "\Downarrow u_{\alpha_X}^{SX,TX}"] & PTX \arrow[r, "PT\varphi"] \arrow[d, phantom, "\Downarrow P\alpha_{\varphi}^{X,Y}"] & PTPY \arrow[r, "P_{\lambda_Y}"] \arrow[d, phantom, "\Downarrow P\alpha^*"] & P^2TY \arrow[r, "m_{TY}"] \arrow[d, phantom, "\Downarrow r_{TY}"] & PTY \\
\arrow[r, phantom, "\Downarrow u_{T\varphi}^{SX,SPY}"] & PSX \arrow[ur, "P\alpha_X"'] \arrow[r, "PS\varphi"] & PSPY \arrow[ur, "P\alpha_{PY}"] \arrow[r, "P\rho_Y"] \arrow[d, phantom, "\Downarrow u_{\rho_Y}^{SPY,PSY}"] & P^2SY \arrow[ur, "P^2\alpha_Y"] \arrow[d, phantom, "\Downarrow u_{P\alpha_Y}^{PSY,PTY}"] & \arrow[d, phantom, "\Downarrow l_{TY}^{-1}"] & \\
& SPY \arrow[ur, "u_{SPY}"] \arrow[r, "\rho_Y"'] & PSY \arrow[ur, "u_{PSY}"] \arrow[r, "P\alpha_Y"'] & PTY \arrow[uur, "u_{PTY}"] \arrow[uurr, "1_{PTY}"] \arrow[r, "Pu_{TY}"'] & P^2TY \arrow[uur, "m_{TY}"'] &
\end{tikzcd}$$

(7.3)

Note that the top row of the diagram is the composition $\tilde{T}\varphi \cdot \beta_X$ in $\mathcal{K}$. The bottom row is $\beta_Y \cdot \tilde{S}\varphi$ expressed as a composition of $\mathcal{K}$ morphisms. We also note that from hereon we shall include less detail in our pasting diagrams. In particular, we shall omit labels on certain 2-cells.

(2) Next, we need to show that for all 0-cells X, Y in $\mathcal{K}$, the pseudo-naturality morphism $\beta_{\varphi}^{X,Y}$ is natural in φ. To do this we need to verify that for each 2-cell

$$\begin{tikzcd}
X \arrow[r, bend left, "\varphi"] \arrow[r, bend right, "\psi"'] \arrow[r, phantom, "\Downarrow\gamma"] & Y
\end{tikzcd}$$

the following diagram commutes:

$$\begin{tikzcd}
\tilde{T}\varphi \cdot \beta_X \arrow[r, "\beta_{\varphi}^{X,Y}"] \arrow[d, "\tilde{S}(\gamma)\cdot\beta_X"'] & \beta_Y \cdot \tilde{S}\varphi \arrow[d, "\beta_Y\cdot\tilde{T}(\gamma)"] \\
\tilde{T}\psi \cdot \beta_X \arrow[r, "\beta_{\psi}^{X,Y}"'] & \beta_Y \cdot \tilde{S}\psi
\end{tikzcd}$$

In the following diagram, the 2-cells should be understood to be the composite 2-cells formed by pasting together the relevant 2-cells from diagram 7.7.

$$\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "u_{TX}"'] \arrow[ddr, bend right, "S\psi"'] & TX \arrow[r, "u_{TX}"] \arrow[d, phantom, "\Downarrow"] & PTX \arrow[r, bend left, "PT\varphi"] \arrow[r, bend right, "PT\psi"'] \arrow[r, phantom, "\Downarrow\tilde{S}(\gamma)"] \arrow[d, phantom, "\Downarrow"] & PTPY \arrow[r, "P_{\lambda_Y}"] & P^2TY \arrow[r, "m_{TY}"] \arrow[d, phantom, "\Downarrow"] & PTY \\
& PSX \arrow[ur, "P\alpha_X"'] \arrow[r, "PT\psi"] \arrow[d, phantom, "\Downarrow"] & PSPY \arrow[ur, "P\alpha_{PY}"'] & & & \\
& SPY \arrow[ur, "u_{SPY}"'] \arrow[r, "\rho_Y"'] & PSY \arrow[r, "P\alpha_Y"'] & PTY \arrow[r, "Pu_{TY}"'] & P^2TY \arrow[uur, "m_{TY}"'] &
\end{tikzcd} \tag{7.4}$$

this diagram equal to

$$\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "u_{TX}"'] \arrow[ddr, bend right, "S\psi"'] & TX \arrow[r, "u_{TX}"] \arrow[d, phantom, "\Downarrow"] & PTX \arrow[r, "PT\varphi"] \arrow[d, phantom, "\Downarrow"] & PTPY \arrow[r, "P_{\lambda_Y}"] & P^2TY \arrow[r, "m_{TY}"] \arrow[d, phantom, "\Downarrow"] & PTY \\
& PSX \arrow[ur, "P\alpha_X"] \arrow[r, bend left, "PS\varphi"] \arrow[r, bend right, "PS\psi"'] \arrow[r, phantom, "\Downarrow\tilde{S}(\gamma)"] \arrow[d, phantom, "\Downarrow"] & PSPY \arrow[ur, "P\alpha_{PY}"'] & & & \\
& SPY \arrow[ur, "u_{SPY}"'] \arrow[r, "\rho_Y"'] & PSY \arrow[r, "P\alpha_Y"'] & PTY \arrow[r, "Pu_{TY}"'] & P^2TY \arrow[uur, "m_{TY}"'] &
\end{tikzcd} \tag{7.5}$$

by the naturality of $\alpha_{X,PY}$ in φ. The last 2-cell is the same as

$$
\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "u_{TX}"] \arrow[ddr, "S\varphi"'{name=A}] \arrow[ddr, bend right=60, "S\psi"'{name=B}] \arrow[from=A, to=B, phantom, "\Swarrow\tilde{S}(\gamma)"] & TX \arrow[r, "u_{TX}"] \arrow[d, phantom, "\Downarrow"] & PTX \arrow[r, "PT\varphi"] \arrow[d, phantom, "\Downarrow"] & PTPY \arrow[r, "P_{\lambda_Y}"] & P^2TY \arrow[r, "m_{TY}"] \arrow[dd, phantom, "\Downarrow"] & PTY \\
 & PSX \arrow[ur, "P\alpha_X"] \arrow[r, "PT\psi"] \arrow[d, phantom, "\Downarrow"] & PSPY \arrow[ur, "P\alpha_{PY}"'] & & & \\
 & SPY \arrow[ur, "u_{SPY}"'] \arrow[r, "\rho_Y"'] & PSY \arrow[r, "P\alpha_Y"'] & PTY \arrow[r, "Pu_{TY}"'] & P^2TY \arrow[uur, "m_{TY}"'] &
\end{tikzcd}
\tag{7.6}
$$

by the naturality of $u^{TX,TPY}$. This concludes the proof of the naturality of $\beta^{X,Y}$.

(3) Finally, we need to verify that the coherence conditions for pseudo-natural transformations hold. We begin with the simpler of the two: we wish to prove that

$$
\begin{tikzcd}
SX \arrow[r, "\beta_X"] \arrow[d, "S(\mathsf{id}_X)"'] \arrow[dr, phantom, "\Downarrow \beta^{X,X}_{\mathsf{id}_X}"] & TX \arrow[d, "T(\mathsf{id}_X)"] \\
SX \arrow[r, "\beta_X"'] & TX
\end{tikzcd}
$$

is the same as the 2-cell id_{β_X}. Actually, since $\mathcal{K}$ is a bicategory, the exact equality we wish to prove is

$$\beta_X \xrightarrow{\simeq} \mathsf{id}_{TX} \cdot \beta_X \xrightarrow{\simeq} \tilde{T}(\mathsf{id}_X) \cdot \beta_X \xrightarrow{\beta^{X,X}_{\mathsf{id}_X}} \beta_X \cdot \tilde{T}(\mathsf{id}_X) \xrightarrow{\simeq} \beta_X \cdot \mathsf{id}_{TX} \xrightarrow{\simeq} \beta_X$$

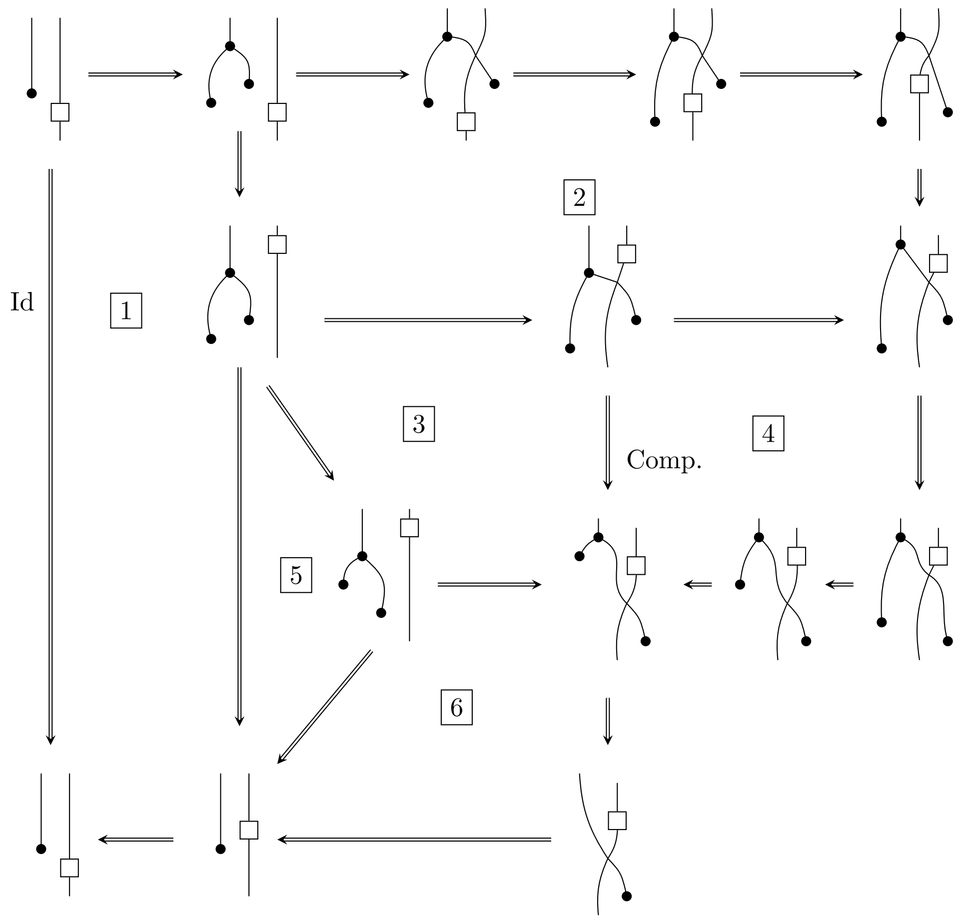
equals

$$\beta_X \xrightarrow{\mathsf{id}_{\beta_X}} \beta_X$$

The pasting diagram in $\mathcal{K}$ for the top 2-cell is given below

$$
\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "u_{SX}"'] \arrow[ddr, bend right, "Su_X"'] \arrow[ddrr, bend right=60, "u_{SX}"'] \arrow[dddr, bend right=70, "\alpha_X"'] & TX \arrow[r, "u_{TX}"] \arrow[d, phantom, "\Downarrow"] & PTX \arrow[r, "PTu_X"] \arrow[rr, bend left, "Pu_{TX}"] \arrow[rrr, bend left=40, "1_{PTX}"] \arrow[d, phantom, "\Downarrow"] & PTPX \arrow[r, "P_{\lambda_X}"] \arrow[d, phantom, "\Downarrow"] & P^2TX \arrow[r, "m_{TX}"] \arrow[d, phantom, "\Downarrow"] & PTX \\
 & PSX \arrow[ur, "P\alpha_X"'] \arrow[r, "PTu_X"] \arrow[d, phantom, "\Downarrow"] & PSPX \arrow[ur, "P\alpha_{PX}"] \arrow[r, "P\rho_X"] \arrow[d, phantom, "\Downarrow"] & P^2SX \arrow[ur, "P^2\alpha_X"] \arrow[d, phantom, "\Downarrow"] & \arrow[d, phantom, "\Downarrow"] & \\
 & SPX \arrow[ur, "u_{SPX}"] \arrow[r, "\rho_X"'] & PSX \arrow[ur, "u_{PSX}"] \arrow[r, "P\alpha_X"'] & PTX \arrow[r, "Pu_{TX}"'] \arrow[uur, "u_{PTX}"'] \arrow[uurr, "1_{PTX}"'] \arrow[uurr, bend right=50, "1_{PTX}"'] & P^2TX \arrow[uur, "m_{TX}"'] & \\
 & TX \arrow[urr, "u_{TX}"'] & & & &
\end{tikzcd}
\tag{7.7}
$$

To show that the above pasting diagram indeed equal to the identity, we need to prove the commutativity of the following diagram given in the graphical language.



We simplify the problem by breaking down the outer diagram into smaller diagrams and prove commutativity for them.

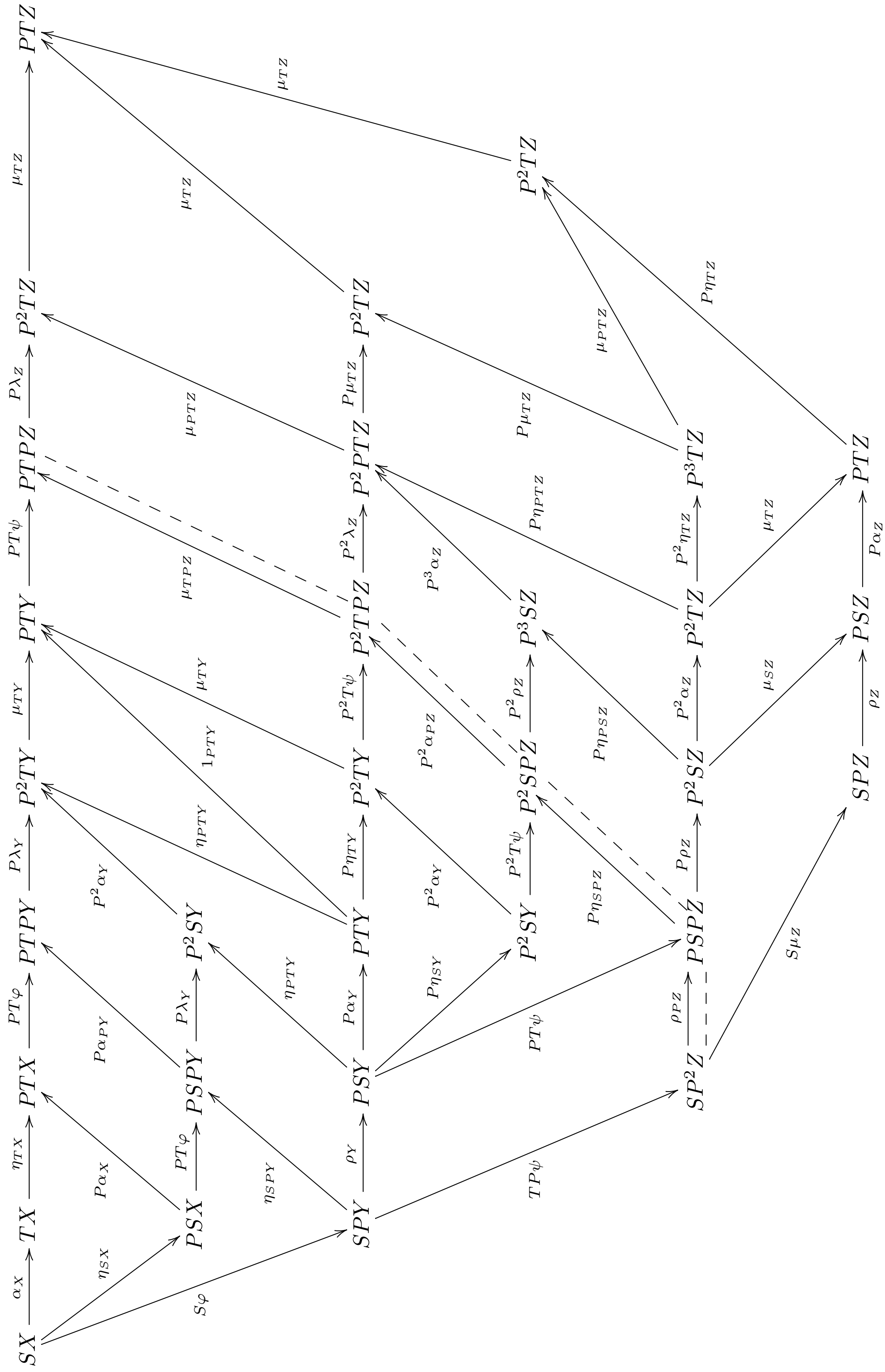
1. This commutativity of this diagram follows from Proposition 4.15.
2. Follows from the coherence conditions for α and several applications of Proposition 4.15.
3. Follow from Proposition 4.15.
4. The vertical arrow is just a composite all the other rewrites.
5. This follows from Proposition 8.1 in [Marmolejo, 1997].
6. Follows from Proposition 4.14.

Finally, to complete the proof, we need to prove the second coherence condition for pseudo-natural transformations, i.e to show the following equality in $\mathsf{kl}(P)$

$$
\begin{tikzcd}
SX \arrow[r, "\beta_X"] \arrow[d, "\tilde{S}\varphi"'] \arrow[dd, bend right=70, "\tilde{T}(\psi\cdot\varphi)"'] \arrow[dd, phantom, bend right=35, "\Downarrow"] \arrow[dr, phantom, "\Downarrow"] & TX \arrow[d, "\tilde{T}\varphi"] \\
SY \arrow[r, "\beta_Y"] \arrow[d, "\tilde{T}(\psi)"'] \arrow[dr, phantom, "\Downarrow"] & TY \arrow[d, "\tilde{T}(\psi)"] \\
SZ \arrow[r, "\beta_Z"'] & TZ
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
SX \arrow[r, "\beta_X"] \arrow[dd, "\tilde{S}(\psi\cdot\varphi)"'] \arrow[ddr, phantom, "\Downarrow"] & TX \arrow[dd, "\tilde{T}(\psi\cdot\varphi)"] \arrow[dr, bend left, "\tilde{T}\varphi"] & \\
 & \arrow[r, phantom, "\Downarrow"] & TY \arrow[dl, bend left, "\tilde{T}(\psi)"] \\
SZ \arrow[r, "\beta_Z"'] & TZ &
\end{tikzcd}
$$

As before, we begin by writing the pasting diagram for each one of the above 2-cells in $\mathcal{K}$. We begin with the left-hand side:

$$
\begin{tikzcd}
SX \arrow[r, "\alpha_X"] \arrow[dr, "\eta_{SX}"'] \arrow[ddr, "T\varphi"'] & TX \arrow[r, "\eta_{TX}"] & PTX \arrow[r, "PT\varphi"] & PTPY \arrow[r, "T\lambda_Y"] \arrow[drr, "PTP\psi"] & P^2TY \arrow[r, "\mu_{TY}"] \arrow[dr, "P^2T\psi"] & PTY \arrow[r, "PT\psi"] & PTPZ \arrow[r, "P\lambda_Z"] \arrow[dl, dash, dashed] & P^2TZ \arrow[rrr, "\mu_{TZ}"] & & & PTZ \\
& PSX \arrow[r, "PT\varphi"] \arrow[ur, "P\alpha_X"] & PSPY \arrow[ur, "P\alpha_{PY}"] \arrow[dr, "PTP\psi"] & & PTP^2Z \arrow[r, "P\lambda_{PZ}"] \arrow[r, dash, dashed, shift right=2] \arrow[dr, "PT\mu_Z"] & P^2TPZ \arrow[ur, "\mu_{TPZ}"] \arrow[r, "P^2\lambda_Z"] & P^3TZ \arrow[ur, "\mu_{TPZ}"] \arrow[rr, "P\mu_{TZ}"] & & P^2TZ \arrow[urr, "\mu_{TZ}"] & P^2TZ \arrow[ur, "\mu_{TZ}"'] & \\
& SPY \arrow[ur, "\eta_{SPY}"] \arrow[dr, "TP\psi"'] & & PSP^2Z \arrow[ur, "P\alpha_{P^2Y}"] \arrow[dr, "PT\mu_Z"'] & & PTPZ \arrow[urrr, "P\lambda_Z"] & & & & & \\
& & SP^2Z \arrow[ur, "\eta_{SP^2Z}"] \arrow[dr, "S\mu_Z"'] \arrow[uurr, dash, dashed] & & PSPZ \arrow[ur, "P\alpha_{PZ}"] \arrow[r, "P\rho_Z"] & P^2SZ \arrow[uurrr, "P^2\alpha_Z"] & PTZ \arrow[uurr, "\eta_{PTZ}"] \arrow[uurrr, "P\eta_{TZ}"'] & & & & \\
& & & SPZ \arrow[r, "\rho_Z"'] \arrow[ur, "\eta_{SPZ}"'] & PSZ \arrow[ur, "\eta_{PSZ}"] \arrow[urr, "P\alpha_Z"'] & & & & & &
\end{tikzcd}
\tag{7.8}
$$



(7.9)

We wish to apply graphical reasoning to show the equality of these two diagrams. But this cannot be done directly, since 1-cells like $T\psi$ for example do not live in $\mathsf{BiCat}(\mathcal{K},\mathcal{K})$. Thus our goal is to divide the diagrams into two parts. One part living in $\mathcal{K}$ and the other a component at a 0-cell Z of a diagram living in $\mathsf{BiCat}(\mathcal{K},\mathcal{K})$. Diagrams 7.8 and 7.9 are already divided into two parts. All the cells on the left side of the dotted line constitute the first part. Cells on the right of the diagrams make up the second part.

We observe that the following sub-diagram of diagram 7.9

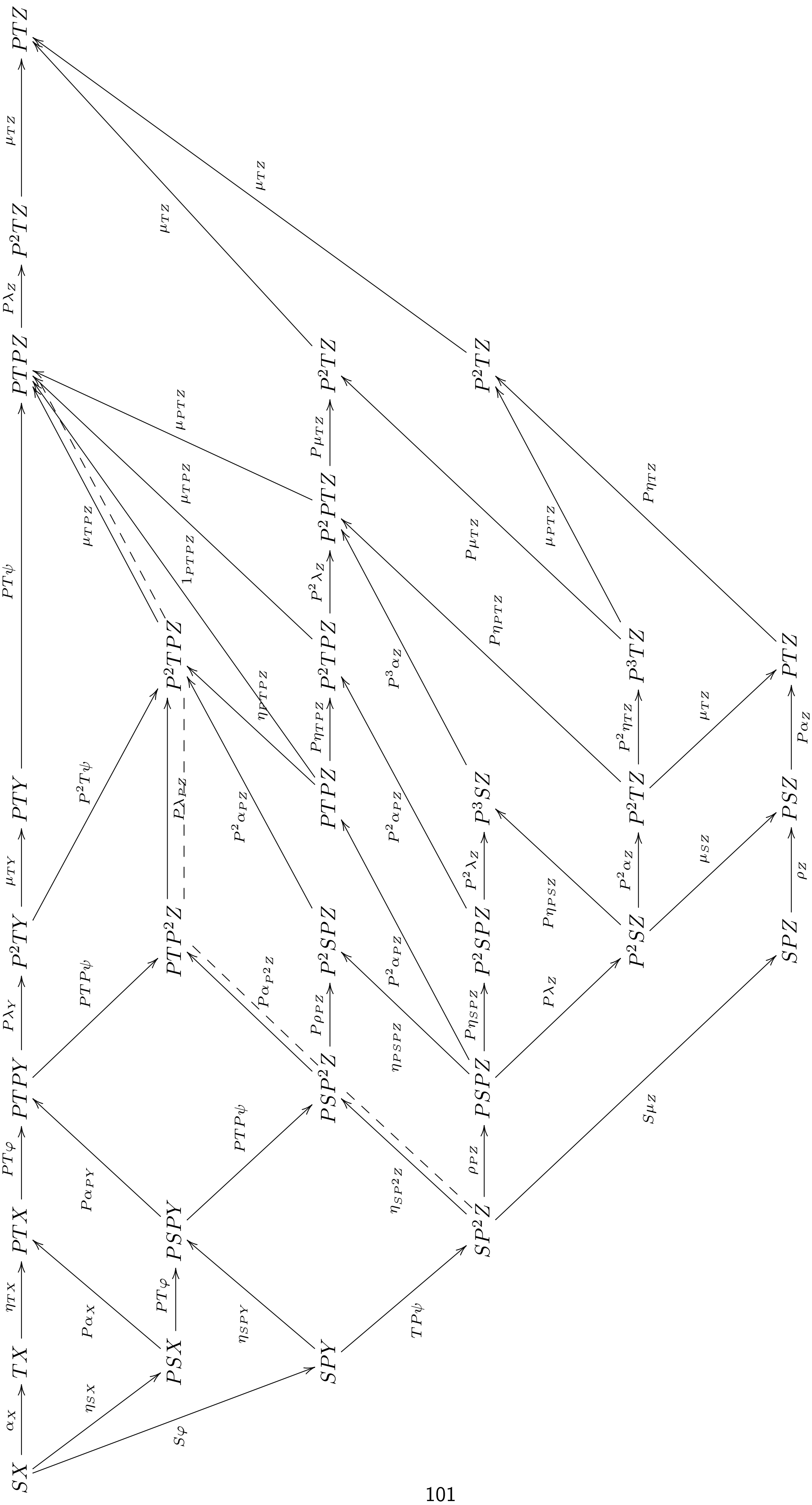
$$
\begin{tikzcd}
 & & PTPY \arrow[r, "P\lambda_Y"] & P^2TY \arrow[r, "\mu_{TY}"] & PTY \arrow[r, "PT\psi"] & PTPZ \\
 & PSPY \arrow[ur, "P\alpha_{PY}"] \arrow[r, "P\lambda_Y"] & P^2SY \arrow[ur, "P^2\alpha_Y"] & & & \\
SPY \arrow[ur, "\eta_{SPY}"] \arrow[r, "\rho_Y"] \arrow[ddr, "TP\psi"'] & PSY \arrow[ur, "\eta_{PTY}"'] \arrow[r, "P\alpha_Y"] \arrow[dr, "P\eta_{SY}"] \arrow[ddr, "PT\psi"'] & PTY \arrow[uur, "\eta_{PTY}"'] \arrow[uurr, "1_{PTY}"'] \arrow[r, "P\eta_{TY}"] & P^2TY \arrow[uur, "\mu_{TY}"'] \arrow[r, "P^2T\psi"] & P^2TPZ \arrow[uur, "\mu_{TPZ}"] & \\
 & & P^2SY \arrow[ur, "P^2\alpha_Y"] \arrow[r, "P^2T\psi"] & P^2SPZ \arrow[ur, "P^2\alpha_{PZ}"] & & \\
 & SP^2Z \arrow[r, "\rho_{PZ}"] & PSPZ \arrow[ur, "P\eta_{SPZ}"] & & &
\end{tikzcd}
\tag{7.10}
$$

equals to

$$
\begin{tikzcd}
 & & PTPY \arrow[r, "T\lambda_Y"] \arrow[dr, "PTP\psi"] & P^2TY \arrow[r, "\mu_{TY}"] \arrow[drr, "P^2T\psi"] & PTY \arrow[rr, "PT\psi"] & & PTPZ \\
 & PSPY \arrow[ur, "P\alpha_{PY}"] \arrow[dr, "PTP\psi"] & & PTP^2Z \arrow[rr, "P\lambda_{PZ}"'] & & P^2TPZ \arrow[ur, "\mu_{TPZ}"] & \\
SPY \arrow[ur, "\eta_{SPY}"] \arrow[dr, "TP\psi"'] & & PSP^2Z \arrow[ur, "P\alpha_{PY}"'] \arrow[r, "P\rho_{PZ}"] & P^2SPZ \arrow[urr, "P\alpha_{PZ}"] & PTPZ \arrow[ur, "\eta_{PTPZ}"'] \arrow[r, "P\eta_{TPZ}"] \arrow[uurr, bend right, "1_{PTPZ}"'] & P^2TPZ \arrow[uur, bend right, "mu_{TPZ}"'] & \\
 & SP^2Z \arrow[ur, "\eta_{SP^2Z}"'] \arrow[r, "\rho_{PZ}"'] & PSPZ \arrow[ur, "\eta_{PSPZ}"'] \arrow[urr, bend right, "P\alpha_{PZ}"'] \arrow[r, "P\eta_{SPZ}"'] & P^2SPZ \arrow[urr, "P^2\alpha_{PZ}"'] & & &
\end{tikzcd}
\tag{7.11}
$$

because all the 2-cells outside the dotted region are modifications. The equality follows after we apply the modification condition six times.

Thus we showed that diagram 7.9 equals to

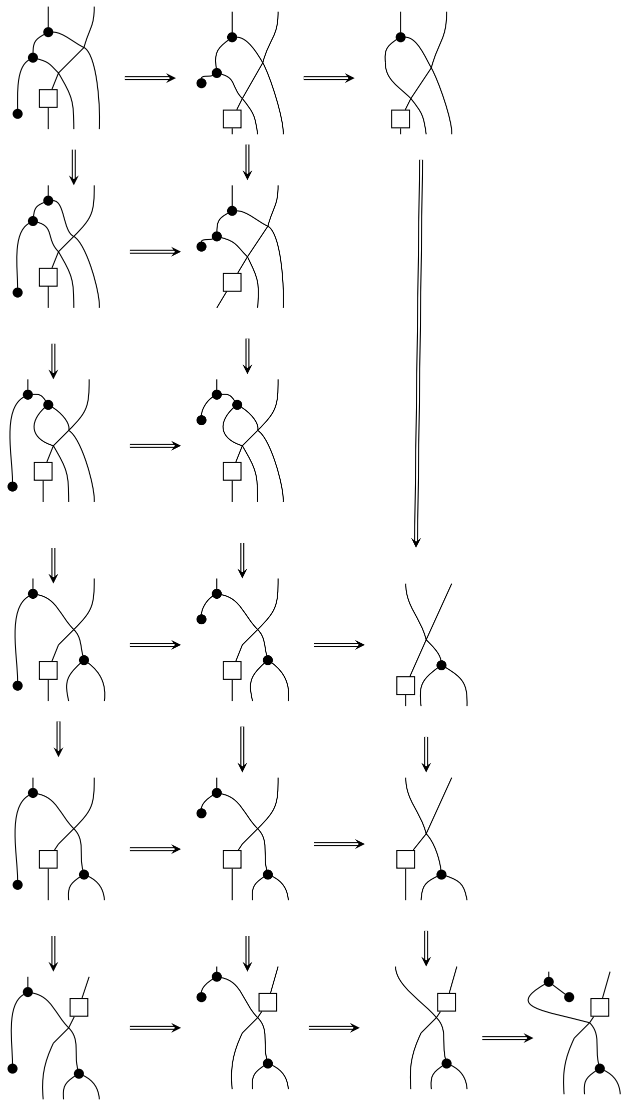

SX
TX
PTX
$PTPY$
P^2TY
PTY
$PTPZ$
P^2TZ
PTZ
α_X
η_{TX}
$PT\varphi$
$P\lambda_Y$
μ_{TY}
$PT\psi$
$P\lambda_Z$
μ_{TZ}
PSX
$PSPY$
PTP^2Z
P^2TPZ
η_{SX}
$P\alpha_X$
$PT\varphi$
$P\alpha_{PY}$
$PTP\psi$
$P^2T\psi$
$P\lambda_{PZ}$
μ_{TPZ}
SPY
PSP^2Z
P^2SPZ
$PTPZ$
P^2TPZ
P^2PTZ
P^2TZ
$S\varphi$
η_{SPY}
$PTP\psi$
$P\alpha_{P^2Z}$
$P\rho_{PZ}$
$P^2\alpha_{PZ}$
$P\eta_{TPZ}$
η_{PTPZ}
1_{PTPZ}
μ_{TPZ}
$P^2\lambda_Z$
μ_{PTZ}
$P\mu_{TZ}$
μ_{TZ}
SP^2Z
$PSPZ$
P^2SPZ
P^3SZ
P^2TZ
$TP\psi$
η_{SP^2Z}
ρ_{PZ}
η_{PSPZ}
$P^2\alpha_{PZ}$
$P\eta_{SPZ}$
$P^2\alpha_{PZ}$
$P^2\lambda_Z$
$P^3\alpha_Z$
$P\eta_{PSZ}$
$P\eta_{PTZ}$
$P\mu_{TZ}$
μ_{TZ}
P^2SZ
P^2TZ
P^3TZ
$S\mu_Z$
$P\lambda_Z$
$P^2\alpha_Z$
$P^2\eta_{TZ}$
μ_{SZ}
μ_{TZ}
μ_{PTZ}
SPZ
PSZ
PTZ
ρ_Z
$P\alpha_Z$
$P\eta_{TZ}$

(7.12)

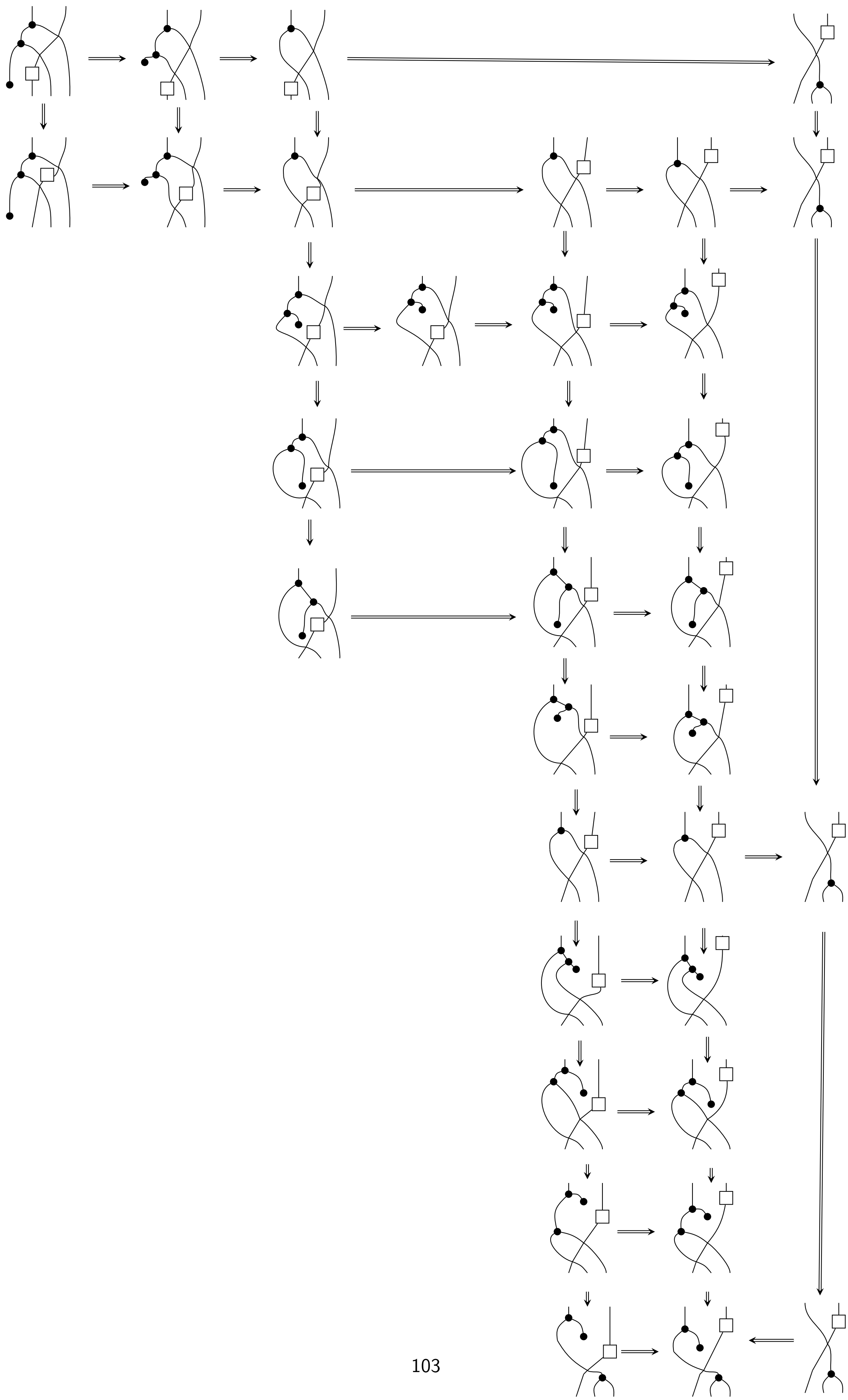
Now observe that sub-diagrams left of the dotted line of both diagram 7.12 and diagram 7.8 are equal.

We proceed with the proof by using graphical reasoning to prove the equality of the sub-diagrams to the right of the dotted line.

The following diagram contains the 2-cell of diagram 7.8 as the bottom-left passage.



And the following diagram contains the 2-cell of digram 7.12 as the bottom-left passage.



We finish the proof by noting that all the inner rectangles in that make-up both diagrams commute and the upper-right passages of these diagrams are equal.

□

7.3.3 2-cells

Theorem 7.6. *Let* $\xi : (\alpha, \alpha^*) \to (\beta, \beta^*)$ *be a 2-cell in* DL_P. *Set*

$$\tilde{\xi} = \begin{tikzcd} SX \arrow[r, bend left, "\alpha"] \arrow[r, bend right, "\beta"'] \arrow[r, phantom, "\Downarrow\xi"] & TX \arrow[r, "u"] & PTX \end{tikzcd} .$$

Then the pair $(\xi, \tilde{\xi})$ *is a 2-cell in* Ext^P.

Proof. We only need to verify the following identity.

$$\begin{tikzcd} SX \arrow[r, bend left, "\alpha_X"] \arrow[r, bend right, "\beta_X"'] \arrow[r, phantom, "\Downarrow\zeta_X"] \arrow[d, "\tilde{S}\varphi"'] & TX \arrow[d, "\tilde{T}\varphi"] \\ SY \arrow[r, "\beta_Y"'] & TY \end{tikzcd} \quad = \quad \begin{tikzcd} SX \arrow[r, "\alpha_X"] \arrow[d, "\tilde{S}\varphi"'] & TX \arrow[d, "\tilde{T}\varphi"] \\ SY \arrow[r, bend left, "\alpha_X"] \arrow[r, bend right, "\beta_X"'] \arrow[r, phantom, "\Downarrow\zeta_X"] & TY \end{tikzcd}$$

The left-hand-side equals:

$$\begin{tikzcd} SX \arrow[r, bend left, "\alpha_X"] \arrow[r, bend right, "\beta_X"'] \arrow[r, phantom, "\Downarrow\zeta_X"] \arrow[dr, "u_{SX}"] \arrow[dd, bend right, "S\varphi"'] & TX \arrow[r, "u_{TX}"] & PTX \arrow[r, "PT\varphi"] & PTPY \arrow[r, "P_{\lambda_Y}"] & P^2TY \arrow[r, "m_{TY}"] & PTY \\ & PSX \arrow[ur, bend left, "P\beta_X"] \arrow[ur, dashed] \arrow[r, "PT\varphi"] \arrow[u, phantom, "(1)"] & PSPY \arrow[ur, bend left, "P\beta_{PY}"] \arrow[ur, dashed] \arrow[r, "P\rho_Y"] \arrow[u, phantom, "(2)"] & P^2SY \arrow[ur, bend left, "P^2\beta_Y"] \arrow[ur, dashed] \arrow[u, phantom, "(3)"] & & \\ & SPY \arrow[u, phantom, ""] \arrow[ur, "u_{SPY}"] \arrow[r, "\rho_Y"'] & PSY \arrow[ur, "u_{PSY}"] \arrow[r, bend left, "P\beta_Y"] \arrow[r, bend right, dashed] & PTY \arrow[u, phantom, "(4)"] \arrow[uur, "u_{PTY}"] \arrow[uurr, "1_{PTY}"] \arrow[r, "Pu_{TY}"'] & P^2TY \arrow[uur, "m_{TY}"'] & \end{tikzcd}$$

We visualise the transformation of this diagram into the right-hand-side of the modification condition via four rewrites, which are depicted on the above diagram with dotted lines. The fact that pre- and post- rewrite diagrams are equal follows from the following observations. (1) and (4) follow from the naturality of u. (2) is the modification condition for ζ and (3) is a consequence of equation **??**. □

7.3.4 Functoriality

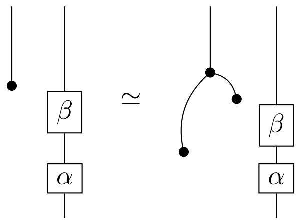
We defined $(\tilde{-})$ on 0-, 1- and 2-cells of DL^P. In this section we show that these assignments define a 2-functor

$$(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P.$$

Proposition 7.7. *Let* $(\alpha, \alpha^*) : (S, \rho) \to (T, \lambda)$, $(\beta, \beta^*) : (T, \lambda) \to (W, \zeta)$ *be 1-cells in* DL^P. *Then there exists an invertible isomorphism*

$$\widetilde{\beta \cdot \alpha} = \tilde{\beta} \cdot \tilde{\alpha}$$

Proof. The result immediately follows from the monad structure of P and the following picture



$\square$

Proposition 7.8. $(\tilde{-})$ *preserves vertical and horizontal composition of 2-cells in* DL^P.

Proof. Since $(\tilde{-})$ acts on 2-cells by whiskering, this is a consequence the 2-functoriality of $(-)_*$. $\square$

Thus we have proved

Theorem 7.9. $(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P$ *is a 2-functor.*

Furthermore, $(\tilde{-})$ preserves the monoidal structure of DL^P.

Proposition 7.10. *Let* (S, ρ) *and* (T, λ) *be 0-cells in* DL^P. *Then*

$$\tilde{S} \otimes \tilde{T} = \widetilde{S \otimes T}$$

Proof. Those functors are clearly equal on 0-cells. On 1- and 2-cells

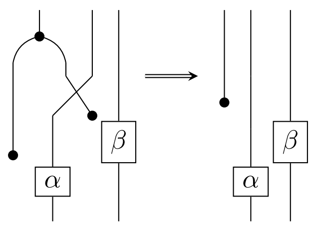
$$\begin{aligned}
\tilde{S} \otimes \tilde{T}(\varphi : X \to PY) &= \tilde{S} \cdot \tilde{T}(\varphi : X \to PY) \\
&= \tilde{S}(\ TX \xrightarrow{T\varphi} TPY \xrightarrow{\rho_Y} PTY\ ) \\
&= \ STX \xrightarrow{ST\varphi} TPY \xrightarrow{S\rho_Y} SPTY \xrightarrow{\lambda_{TY}} PSTY \\
&= \widetilde{S \otimes T}(\varphi : X \to PY)
\end{aligned}$$

$\square$

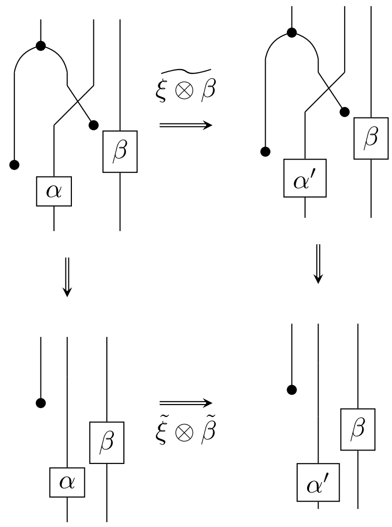
Proposition 7.11. *Let* $(\alpha, \alpha^*) : (S, \rho) \to (S', \rho')$ *and* $(\beta, \beta^*) : (T, \lambda) \to (T', \lambda')$ *be 1-cells in* DL^P. *Then there a natural isomorphism*

$$\gamma_{\alpha,\beta} : \tilde{\alpha} \otimes \tilde{\beta} \to \widetilde{\alpha \otimes \beta}$$

Proof. The modification $\gamma_{\alpha,\beta}$ is given by



Naturality of γ is α means that for every 2-cell $\xi : \alpha \to \alpha'$ in DL^P, the following diagram commutes:



This diagram commutes because all the rewrites that constitute $\gamma_{\alpha,\beta}$ are also natural.

□

Proposition 7.12. *Let ξ and ξ' be 2-cells in DL^P. Then*

$$\widetilde{\xi \otimes \xi'} = \tilde{\xi} \otimes \tilde{\xi}'$$

Proof. This follows from one application of the naturally of γ in α and one applications of its naturally in β. (See the proof of Proposition 7.11 for the definition of γ). □

Putting all our results together, we conclude

Theorem 7.13. $(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P$ *is a monoidal 2-functor.*

7.4 From extensions to 2-distributive laws

In this section we describe the passage from the 2-category Ext_P to DL^P.

We note that the results of this subsection are not used in the rest of this thesis; we add them here for completeness. We also note that for this reason we omitted many details from the proofs of our claims.

7.4.1 0-cells

Since the equivalence of 2-distributive laws and extensions is well known, we only sketch the proof of the following result.

Theorem 7.14. *Let $(S, \tilde{S})$ be a 0-cell in Ext_P. Define a 2-natural transformation*

$$\rho = \ SP \xrightarrow{\tilde{S}1_P} PS$$

Then the pair (S, ρ) is a 0-cell in DL_P.

Note that here $1_P : P \to P$ is viewed as a 1-cell $P \nrightarrow 1$ in $\mathsf{kl}(P)$. Then $\tilde{S}(1_P) : SP \nrightarrow S$ in $\mathsf{kl}(P)$ and hence is a 1-cell $SP \to PS$ in $\mathcal{K}$.

Sketch of proof. We begin by showing how to construct the pseudo-naturality of $\tilde{S}(1_P)$. Let $f : X \to Y$ be a morphism in $\mathcal{K}$. We want to show the existence of the following isomorphism

$$\begin{tikzcd}
SPX \arrow[r, "\tilde{S}(1_P)_X"] \arrow[d, "SPf"'] \arrow[dr, phantom, "\Downarrow"] & PSX \arrow[d, "PSf"] \\
SPY \arrow[r, "\tilde{S}(1_P)_Y"'] & PSY
\end{tikzcd}$$

Lets begin with examining the right-hand-side.

$$\begin{aligned}
PSf \cdot \tilde{S}(1_P)_X &\simeq SPX \xrightarrow{\tilde{S}(1_P)_X} SX \xrightarrow{(Sf)_*} SY \\
&\simeq SPX \xrightarrow{\tilde{S}(1_P)_X} SX \xrightarrow{\tilde{S}(f_*)} SY \\
&\simeq SPX \xrightarrow{\tilde{S}(f_* \cdot 1_P)_X} SY
\end{aligned}$$

where the first isomorphism holds since the last two terms of the composite $SPX \xrightarrow{\tilde{S}(1_P)_X} SX \xrightarrow{(Sf)_*} SY$ in $\mathcal{K}$ are isomorphic to the identity.

Finally, note that the $(f_* \cdot 1_P)_X = (f_* \cdot 1_{PX}) \simeq Pf$. In detail,

$$\begin{tikzcd}
PX \arrow[r, "1_{PX}"] \arrow[rrrr, bend right=40, "Pf"'] \arrow[rrrr, phantom, bend right=25, "\Downarrow"] & PX \arrow[r, "Pf"] & PY \arrow[r, "u_{PY}"] \arrow[rr, bend right, "1_{PY}"'] \arrow[rr, phantom, bend right=15, "\Downarrow"] & P^2Y \arrow[r, "m_Y"] & PY
\end{tikzcd}$$

So, we showed that

$$PSf \cdot \tilde{S}(1_P)_X \simeq \tilde{S}(Pf).$$

Next we look at the left-hand-side. Lemma 7.21 gives us

$$PX \xrightarrow{Pf} Y \;\simeq\; PX \xrightarrow{Pf_*} PY \xrightarrow{1_{PY}} Y$$

hence

$$\begin{aligned}
\tilde{S}(Pf) &\simeq \tilde{S}(1_{PY} \cdot Pf_*) \\
&\simeq \tilde{S}(1_{PY}) \cdot \tilde{S}(Pf_*) \\
&\simeq \tilde{S}(1_{PY}) \cdot S(Pf)_*.
\end{aligned}$$

And the last line is isomorphic to the left-hand-side:

$$\begin{tikzcd} SPX \arrow[r, "SPf"] & SPY \arrow[r, "u_{SPY}"] \arrow[dr, "\tilde{S}(1_{PY})"'] & PSPY \arrow[d, phantom, "\Downarrow"] \arrow[r, "P\tilde{S}(1_{PY})"] & P^2SPY \arrow[r, "m_{SPY}"] \arrow[d, phantom, "\Downarrow"] & PSPY \\ & & PSPY \arrow[ur, "u_{PSPY}"'] \arrow[urr, bend right, "1_{PSPY}"'] & & \end{tikzcd}$$

The next step is to show that $\tilde{S}(1_P)$ is a 2-distributive law. We begin with exhibiting the following isomorphism

$$\begin{tikzcd} SP^2 \arrow[r, "\tilde{S}(1_P)P"] \arrow[d, "Sm"'] & PSP \arrow[r, "P\tilde{S}(1_P)"] \arrow[d, phantom, "\Downarrow"] & P^2S \arrow[d, "mS"] \\ SP \arrow[rr, "\tilde{S}(1_P)"'] & & PS \end{tikzcd}$$

We again analyse the right- and left-hand sides, beginning with the former.

Note

$$\begin{aligned} m_{SX} \cdot P\tilde{S}(1_P)_X \cdot \tilde{S}(1_P)_{PX} \quad &\simeq \quad SP^2X \xrightarrow{\tilde{S}(1_{P^2X})} SPX \xrightarrow{\tilde{S}(1_{PX})} SX \\ &\simeq \quad \tilde{S}(\ P^2X \xrightarrow{1_{P^2X}} PX \xrightarrow{1_{PX}} X\ ) \\ &\simeq \quad \tilde{S}(m_X) \end{aligned}$$

where the last isomorphism follows by computing the composite in $\mathcal{K}$.

Now invoke Lemma 7.21 to see that

$$P^2X \xrightarrow{m_X} X \ \simeq\ P^2X \xrightarrow{(m_X)_*} PX \xrightarrow{1_{PX}} X\ .$$

Hence

$$\begin{aligned} \tilde{S}(m_X) \quad &\simeq \quad \tilde{S}(1_{PX} \cdot (m_X)_*) \\ &\simeq \quad \tilde{S}(1_{PX}) \cdot \tilde{S}((m_X)_*) \\ &\simeq \quad \tilde{S}(1_{PX}) \cdot (Sm_X)_*. \end{aligned}$$

We compute the composite on the last line in $\mathcal{K}$:

$$\begin{tikzcd} SP^2X \arrow[r, "Sm"] & SPX \arrow[r, "u_{SPX}"] \arrow[dr, "\tilde{S}(1_{PX})"'] & PSPX \arrow[d, phantom, "\Downarrow"] \arrow[r, "P\tilde{S}(1_{PX})"] & P^2SX \arrow[r, "m_{SX}"] \arrow[d, phantom, "\Downarrow"] & PSX \\ & & PSX \arrow[ur, "u_{PSX}"'] \arrow[urr, bend right, "1_{PSX}"'] & & \end{tikzcd}$$

which gives the left-hand-side.

Finally, we exhibit the isomorphism

$$\begin{tikzcd} S \arrow[r, "Su"] \arrow[dr, "uS"'] & SP \arrow[d, "\tilde{S}(1_P)"] \\ & PS \end{tikzcd}$$

$$\begin{aligned} \tilde{S}(1_P) \cdot Su &\simeq \tilde{S}(1_P) \cdot S(u)_* \\ &\simeq \tilde{S}(1_P \cdot u_*) \\ &\simeq \tilde{S}(u) \simeq uS \end{aligned}$$

□

7.4.2 1-cells

Theorem 7.15. *Let $(\alpha, \tilde{\alpha})$ be a 1-cell in Ext_P. Then there exists*

$$\alpha^* : P\alpha \cdot \tilde{S}1_P \to \tilde{T}1_P \cdot \alpha P$$

making (α, α^*) a 1-cell in DL^P.

Proof. We produce α^* via the following diagram

$$\begin{tikzcd} SP \arrow[rrrr, "\tilde{S}1_P"] \arrow[dr, "\tilde{\alpha}P"] \arrow[ddd, "\alpha P"'] & & \arrow[d, phantom, "(1)"] & & PS \arrow[dl, "P\tilde{\alpha}"'] \arrow[d, "P\alpha"] \\ & PTP \arrow[r, "P\tilde{T}1_P"] & P^2T \arrow[ddrr, "mT"] & P^2T \arrow[ddr, "m_T"] & PT \arrow[l, "u_{PT}"'] \arrow[dd, "1_{PT}"] \\ & PT \arrow[ur, "u_{PT}"] \arrow[drrr, "1_{PT}"] & & & \\ TP \arrow[uur, "u_{TP}"] \arrow[ur, "\tilde{T}1_P"] \arrow[rrrr, "\tilde{T}1_P"'] & & & & PT \end{tikzcd}$$

where the 2-cell (1) is given by the pseudo-naturality of $\tilde{\alpha}$. □

7.4.3 2-cells

Theorem 7.16. *Let $(\xi, \tilde{\xi}) : (\alpha, \tilde{\alpha}) \to (\beta, \tilde{\beta})$ be a 2-cell in Ext_P. Then $\xi : \alpha \to \beta$ is a 2-cell in DL^P.*

Proof. We need to show

$$\begin{tikzcd} SP \arrow[r, "\tilde{S}1_P"] \arrow[d, bend right=40, "\alpha P"'] \arrow[d, bend left=40, "\beta P"] \arrow[d, phantom, "\overset{\xi P}{\Longrightarrow}"] & PS \arrow[d, "P\beta"] \\ TP \arrow[r, "\tilde{T}1_P"'] & PT \end{tikzcd} = \begin{tikzcd} SP \arrow[r, "\tilde{S}1_P"] \arrow[d, "\alpha P"'] & PS \arrow[d, bend right=40, "P\alpha"'] \arrow[d, bend left=40, "P\beta"] \arrow[d, phantom, "\overset{P\xi}{\Longrightarrow}"] \\ TP \arrow[r, "\tilde{T}1_P"'] & PT \end{tikzcd}$$

We know from condition 7.2 that

$$\tilde{\xi} = uT * \xi;$$

i.e. $\tilde{\xi}$ acts by whiskering. Thus the fact that $\tilde{\xi}$ is a modification translates to

$$\begin{tikzcd}
SP \arrow[rrrrr, "\tilde{S}1_P"] \arrow[dr, "\tilde{\alpha}P"] \arrow[ddd, "\alpha P"'] & & & & & PS \arrow[dl, bend right, "P\tilde{\alpha}"'] \arrow[d, bend right, "P\alpha"'] \arrow[d, bend left, "P\beta"] \arrow[d, phantom, "\overset{P\xi}{\Longrightarrow}"] \\
& PTP \arrow[rr, "P\tilde{T}1_P"] & & P^2T \arrow[ddrr, "mT"] & P^2T \arrow[ddr, "m_T"] & PT \arrow[l, "u_{PT}"'] \arrow[dd, "1_{PT}"] \\
& & PT \arrow[ur, "u_{PT}"] \arrow[drrr, "1_{PT}"] & & & \\
TP \arrow[uur, "u_{TP}"] \arrow[urr, "\tilde{T}1_P"] \arrow[rrrrr, "\tilde{T}1_P"'] & & & & & PT
\end{tikzcd}
=
\begin{tikzcd}
SP \arrow[rrrrr, "\tilde{S}1_P"] \arrow[dr, "\tilde{\beta}P"] \arrow[ddd, bend right, "\alpha P"'] \arrow[ddd, bend left, "\beta P"] \arrow[ddd, phantom, "\overset{\xi P}{\Longrightarrow}"] & & & & & PS \arrow[dl, "P\tilde{\beta}"'] \arrow[d, "P\beta"] \\
& PTP \arrow[rr, "P\tilde{T}1_P"] & & P^2T \arrow[ddrr, "mT"] & P^2T \arrow[ddr, "m_T"] & PT \arrow[l, "u_{PT}"'] \arrow[dd, "1_{PT}"] \\
& & PT \arrow[ur, "u_{PT}"] \arrow[drrr, "1_{PT}"] & & & \\
TP \arrow[uur, "u_{TP}"] \arrow[urr, "\tilde{T}1_P"] \arrow[rrrrr, "\tilde{T}1_P"'] & & & & & PT
\end{tikzcd}$$

which is exactly the desired equality. $\square$

7.4.4 Functoriality

In this section we show that the above constructions on 0-, 1-, and 2-cells define a 2-functor

$$d : \mathsf{Ext}_P \to \mathsf{DL}^P.$$

Proposition 7.17. *Let $\alpha = (\alpha, \tilde{\alpha}) : (S, \tilde{S}) \to (T, \tilde{T})$ and $\beta = (\beta, \tilde{\beta}) : (T, \tilde{T}) \to (W, \tilde{W})$ be composible 1-cells in Ext_P. Then there is a natural isomorphism*

$$d(\beta \cdot \alpha) \simeq d(\beta) \cdot d(\alpha). \tag{7.13}$$

Proof. Consider the following diagram

$$\begin{tikzcd}
SP \arrow[r, "\tilde{S}1_P"] \arrow[d, "\alpha P"'] \arrow[dd, bend right=60, "(\beta\cdot\alpha)P"'] \arrow[dr, phantom, "\Downarrow"] & PS \arrow[d, "P\alpha"] \arrow[dd, bend left=60, "P(\beta\cdot\alpha)"] \\
TP \arrow[r, "\tilde{T}1_P"] \arrow[d, "\beta P"'] \arrow[dr, phantom, "\Downarrow"] & PT \arrow[d, "P\beta"] \\
WP \arrow[r, "\tilde{W}1_P"'] & PW
\end{tikzcd}$$

The left-hand-side of Equation 7.13 is obtained by composing the inner rectangle. The right-hand-side, by composing the outer rectangle. Clearly, the two passages are indeed isomorphic.

Coherence follows from the coherence of the monad P. $\square$

d sends a 2-cell $(\xi, \tilde{\xi})$ in Ext_P to ξ. Hence vertical and horizontal composition of 2-cells is preserved on the nose.

Putting these results together, we get

Theorem 7.18. *$d : \mathsf{Ext}_P \to \mathsf{DL}^P$ is a 2-functor.*

Furthermore, d preserves the monoidal structure of Ext_P. We begin with 0-cells.

Proposition 7.19. *Let $(S, \tilde{S})$ and $(T, \tilde{T})$ be 0-cells in Ext_P. Then*

$$\tilde{S}\tilde{T}(1_{PX}) \simeq \tilde{S}(1_{PTX}) \cdot S(\tilde{T}(1_{PX})).$$

We shall need several facts to complete the proof. Recall from Chapter 5 that there is an adjunction

$$(-)_* \dashv G : \mathsf{kl}(P) \to \mathcal{K}$$

Lemma 7.20. *For any $X \overset{\varphi}{\nrightarrow} Y$ in $\mathsf{kl}(P)$,*

$$X \overset{\varphi_*}{\nrightarrow} PY \;\simeq\; X \overset{\varphi}{\nrightarrow} Y \overset{(u_Y)_*}{\nrightarrow} PX$$

Proof. The right-hand-side is the top line in

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & PY \arrow[rr, "P((u_Y)_*)"] \arrow[dr, "Pu_Y"] \arrow[ddr, "1_{PY}"'] & & P^3Y \arrow[r, "m_{PY}"] & P^2Y \\
& & P^2Y \arrow[ur, "Pu_{P^2Y}"] \arrow[urr, "1_{P^2Y}"'] & & \\
& & PY \arrow[uurr, "u_{PY}"'] & &
\end{tikzcd}$$

and the left-hand-side is the bottom line. $\square$

Lemma 7.21. *For any $\varphi : X \nrightarrow Y$ in $\mathsf{kl}(P)$,*

$$X \overset{\varphi_*}{\nrightarrow} PY \overset{1_{PY}}{\nrightarrow} Y \;\simeq\; X \overset{\varphi}{\nrightarrow} Y$$

Proof. Consider the diagram

$$\begin{tikzcd}
X \arrow[rr, "\varphi_*", "\shortmid"{marking}] \arrow[dr, "\varphi", "\shortmid"{marking}] \arrow[rrr, bend right=90, "\varphi"', "\shortmid"{marking}] & \arrow[d, phantom, "(1)"] & PY \arrow[r, "1_{PY}", "\shortmid"{marking}] \arrow[d, phantom, "(2)"] & Y \\
& Y \arrow[ur, "(u_Y)_*"', "\shortmid"{marking}] \arrow[urr, bend right, "u_Y"', "\shortmid"{marking}] & &
\end{tikzcd}$$

The isomorphism (1) comes from Lemma 7.20. Isomorphisms (2) comes from Equation 5.1 in chapter **??**. $\square$

We can now take $\varphi = \tilde{T}(1_{PX})$ in the previous Lemma to obtain the isomorphism

$$TPX \xrightarrow{\left(\tilde{T}(1_{PX})\right)_*} PTX \xrightarrow{1_{PTX}} TX \;\simeq\; TPX \xrightarrow{\tilde{T}(1_{PX})} TX \tag{7.14}$$

Finally, we prove Proposition 7.19

Proof of Proposition 7.19. Apply $\tilde{S}$ to equation 7.14

$$\begin{aligned} \tilde{S}\tilde{T}(1_{PX}) &\simeq \tilde{S}(1_{PTX}) \cdot \tilde{S}(\tilde{T}(1_{PX})_*) \\ &= \tilde{S}(1_{PTX}) \cdot \left(S(\tilde{T}(1_{PX}))\right)_*. \end{aligned}$$

Finally, notice the isomorphism

$$\begin{tikzcd}
STPX \arrow[r, "S(\tilde{T}(1_{PX}))"] & SPTS \arrow[r, "u_{SPTX}"] \arrow[dr, "\tilde{S}(1_{PTX})"'] & PSPTX \arrow[r, "P\tilde{S}(1_{PTX})"] \arrow[d, phantom, "\Downarrow"] & P^2STX \arrow[r, "m_{STX}"] \arrow[d, phantom, "\Downarrow"] & PSTX \\
 & & PSTX \arrow[ur, "u_{PSTX}"] \arrow[urr, bend right, "1_{PSTX}"'] & &
\end{tikzcd}$$

$\square$

We move on to study the preservation of the tensor on 1-cells.

Proposition 7.22. *Let* $(\alpha, \tilde{\alpha}) : (S, \tilde{S}) \to (T, \tilde{T})$ *and* $(\beta, \tilde{\beta}) : (R, \tilde{R}) \to (W, \tilde{W})$ *be 1-cells in* Ext_P. *Then*

$$\begin{aligned} d\big((\alpha, \tilde{\alpha}) \otimes (\beta, \tilde{\beta})\big) &= d\big((\alpha \otimes \beta, \tilde{\alpha} \otimes \tilde{\beta}\big) \\ &= (\alpha \otimes \beta, (\alpha \otimes \beta)^*) \end{aligned}$$

Sketch of proof. Both the right and the left hand side are modifications of the form

$$P(\alpha \otimes \beta) \cdot \tilde{S}\tilde{T}(1_P) \to \tilde{T}\tilde{W}(1_P) \cdot (\alpha \otimes \beta)P.$$

Both are created by tensoring and composing structural 2-cells for the monad P, the distributive laws $\tilde{S}(1_P)$, $\tilde{T}(1_P)$, $\tilde{R}(1_P)$, $\tilde{R}(1_P)$ and 1-cell $d(\alpha, \tilde{\alpha})$, $d(\beta, \tilde{\beta})$ in DL^P. The coherence axioms for all those structures show that any two such passages from the same domain and codomain must be equal, leading to the desired conclusion. $\square$

Finally,

Proposition 7.23. d *preserves the tensor on 2-cells.*

Proof. This is clear, since d sends a pair $(\xi, \tilde{\xi})$ to ξ. $\square$

Thus we conclude

Theorem 7.24. d *is a monoidal 2-functor.*

7.5 The equivalence

In this section we show that the monoidal 2-functors $d : \mathsf{Ext}_P \to \mathsf{DL}^P$ and $(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P$ form an equivalence of 2-categories.

Theorem 7.25. *The pair $d : \mathsf{Ext}_P \to \mathsf{DL}^P$ and $(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P$ forms an equivalence of 2-categories.*

Proof. We exhibit two 2-natural isomorphism

$$\eta : 1_{\mathsf{DL}_P} \to d \cdot (\tilde{-})$$

and

$$\epsilon : (\tilde{-}) \cdot d \to 1_{\mathsf{Ext}_P}$$

Let $(S, \rho) \in \mathsf{DL}_P$. First note that $\tilde{S}(1_P) = \rho$. Indeed,

$$\tilde{S}(1_P) = \; SP \xrightarrow{1_P} SP \xrightarrow{\rho} PS \; .$$

Thus, we can set

$$\eta_\rho = (1_S, \mathsf{id}) : (S, \rho) \to (S, \tilde{S}(1_P))$$

Clearly η is a 2-natural isomorphism.

It remains to define ϵ. Take $(S, \tilde{S})$ in Ext_P, write $(S, \dot{S}) := \widetilde{d(S, \tilde{S})}$ and set

$$\epsilon_{\tilde{S}} = (1_S, uS).$$

Clearly, this is an isomorphism and uS is a lifting of 1_S. We just need to show that uS is indeed a 2-natural transformation from $\dot{S}$ to $\tilde{S}$. I.e. we need to show that for every $\varphi : X \nrightarrow Y$ in $\mathsf{kl}(P)$

$$\begin{tikzcd}
SX \arrow[r, "u_{SX}"] \arrow[d, "S\varphi"] \arrow[dd, bend right=60, "\dot{S}\varphi"'] & SX \arrow[dd, "\tilde{S}\varphi"] \\
SPY \arrow[d, "\tilde{S}(1_{PY})"] & \\
SY \arrow[r, "u_{SY}"'] & SY
\end{tikzcd}$$

commutes. But this amounts to showing that the following are isomorphic:

$$\tilde{S}(1_{PY}) \cdot S\varphi \simeq \tilde{S}(\varphi).$$

The latter isomorphism is easily derived from Lemma 7.21 as follows

$$\begin{aligned}
\tilde{S}\varphi &\simeq \tilde{S}(1_{PY} \cdot \varphi_*) \\
&\simeq \tilde{S} \cdot \tilde{S}(\varphi_*) \\
&\simeq \tilde{S}(1_{PY}) \cdot (S\varphi)_* \\
&\simeq \tilde{S}(1_{PY}) \cdot S\varphi.
\end{aligned}$$

□

7.6 Extensions of monads

In the previous section we showed that

$$(\tilde{-}) : \mathsf{Ext}^P \to \mathsf{End}(\mathsf{kl}(P))$$

is a monoidal 2-functor. We know that any strong monoidal functor

$$M : X \to Y$$

from a monoidal category X to a monoidal category Y preserves "monoidal" structures in X, such as for example monoids. Similarly, a weak monoidal functor preserves such structure up to isomorphisms. For example, given a monoid $x \otimes x \to x \in X$, we set

$$m' : Mx \otimes Mx \simeq M(x \otimes x) \xrightarrow{Mm} Mx$$

These results can be categorified. *monoidal 2-functors send pseudomonoids to pseudomonoids and pseudomonoid morphism to pseudomonoid morphisms*. The interested reader can find more details on these facts in [McCrudden, 2000], with an accessible treatment of pseudomonoids in [Houston, 2008].

Might need a better reference here. Ask Bob

Further note that *distributive laws of monads over monads are monads in* DL^P. More details on can be found in [Tanaka, 2005].

We translate these results into a Theorem.

Theorem 7.26. *The functor* $(\tilde{-}) : \mathsf{DL}^P \to \mathsf{Ext}_P$ *sends distributive laws of monads over monads to monads in* Ext*.*

Furthermore,

Theorem 7.27. *Let* (M, λ) *and* (S, ρ) *be monoids in* DL^P *and let* $(\alpha, \alpha^*) : (M, \lambda) \to (S, \rho)$ *be a monoid morphism. Then*

$$\tilde{\alpha} : \tilde{M} \to \tilde{S}$$

is a monoid morphism in Ext^P*.*

Chapter 8

Functors between Kleisli bicategories

In this chapter we prove two theorems. The first one shows that every 2-monad morphism between two 2-monads gives rise to a 2-functor between the Kleisli bicategories of these monads. When combined with results from Chapter 7, this theorem shows that certain 2-natural transformation between types of theories give rise to functors between categories of theories of these types. More precisely, every monad morphism $i : S \to T$ in DL^P gives rise to a 2-functor

$$I : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T})$$

For the second theorem, we fix P to be the presheaf monad on Cat. Now suppose we are given a 1-cell $i : S \to T$ in DL^P. At each category X, the component i_X gives rise to two morphisms in Prof:

$$(i_X)_* : SX \nrightarrow TX \text{ and } (i_X)^* : TX \nrightarrow SX$$

In Chapter 7 we proved that the collection $\{(i_X)_* \mid X \in \mathsf{Cat}\ \}$ is actually a 2-natural transformation. We suppose that $i^* = \{(i_X)^* \mid X \in \mathsf{Cat}\}$ is also a 2-natural transformation. In fact we assume that it is a monad morphism. This allows us to apply the first Theorem of this chapter to $i^* : \tilde{T} \to \tilde{S}$ to create a 2-functor $J : \mathsf{kl}(\tilde{T}) \to \mathsf{kl}(\tilde{S})$.

The aim of the second Theorem is to study the relationship between I and J. We show that the adjunction $(i_X)^* \dashv (i_X)_*$ generates a lax-adjunction between I and J.

8.1 Theorem 1

As we mentioned, the first theorem in this chapter shows that 2-monad morphisms give rise to 2-functors between the Kleisli categories of these monads.

Theorem 8.1. *Let $\mathcal{K}$ be a bicategory. Let $S, T : \mathcal{K} \to \mathcal{K}$ be pseudo-monads. Let $(\alpha, \bar{\alpha}, \hat{\alpha}) : S \to T$ be a pseudo-monad morphism. Then the assignment*

$$A : \mathsf{kl}(S) \to \mathsf{kl}(T)$$

given by

$$A(\ X \xrightarrow{\varphi} SY\ ) =\ X \xrightarrow{\varphi} SY \xrightarrow{\alpha_Y} TY$$

gives rise to a pseudo-functor.

Proof. A is identity on 0-cells and is given on 2-cells by whiskering.

For any X, Y, $A : \mathsf{kl}(S)(X, Y) \to \mathsf{kl}(T)(X, Y)$ is a functor. Indeed, given 2-cells $\delta : \varphi \to \psi$, $\gamma : \psi \to \xi$ we have

$$\begin{aligned}
A(\gamma \cdot \delta) &= (\gamma \cdot \delta) * 1_{\alpha_Y} \\
&= (\gamma \cdot \delta) * (1_{\alpha_Y} \cdot 1_{\alpha_Y}) \\
&= (\gamma * 1_{\alpha_Y}) \cdot (\delta * 1_{\alpha_Y}) \\
&= A\gamma \cdot A\delta.
\end{aligned}$$

Next, given 1-cells $\varphi : X \to Y, \psi : Y \to Z$ in $\mathsf{kl}(S)$, we define the isomorphism

$$h^{\varphi,\psi} : A\varphi \cdot A\psi \simeq A(\varphi \cdot \psi)$$

as the following composite 2-cell in $\mathcal{K}$:

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, bend right, "S\psi"'] & TY \arrow[r, "T\psi"] & TSZ \arrow[r, "T\alpha_Z"] & TTZ \arrow[r, "m_Z"] & TZ \\
& & SSZ \arrow[ur, "\alpha_{SZ}"] \arrow[r, "S\alpha_Z"'] \arrow[dr, bend right, "m_Z"'] & STZ \arrow[ur, "\alpha_{TZ}"] & & \\
& & & SZ \arrow[uurr, bend right, "\alpha_Z"'] & &
\end{tikzcd}$$

The naturality of h in both φ and ψ amounts to the commutativity of

$$\begin{tikzcd}
A\psi \cdot A\varphi \arrow[r, "h^{\varphi,\psi}"] \arrow[d, "A\delta * A\gamma"'] & A(\psi \cdot \varphi) \arrow[d, "A(\delta * \gamma)"] \\
A\psi' \cdot A\varphi' \arrow[r, "h^{\varphi',\psi'}"'] & A(\psi' \cdot \varphi')
\end{tikzcd}$$

in $\mathsf{kl}(T)$.

This commutativity is equivalent to showing the equality of

$$\begin{tikzcd}
X \arrow[r, bend left, "\varphi"] \arrow[r, bend right, "\varphi'"'] \arrow[r, phantom, "\Downarrow\gamma"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, bend right, "S\psi'"'] & TY \arrow[r, bend left, "T\psi"] \arrow[r, bend right, "T\psi'"'] \arrow[r, phantom, "\Downarrow T\delta"] & TSZ \arrow[r, "T\alpha_Z"] & TTZ \arrow[r, "m_Z"] & TZ \\
& & SSZ \arrow[ur, bend right, "\alpha_{SZ}"] \arrow[r, "S\alpha_Z"'] \arrow[dr, bend right, "m_Z"'] & STZ \arrow[ur, "\alpha_{TZ}"'] & & \\
& & & SZ \arrow[uurr, bend right, "\alpha_Z"'] & &
\end{tikzcd}$$

and

$$\begin{tikzcd}
X \arrow[r, bend left, "\varphi"] \arrow[r, bend right, "\varphi'"'] \arrow[r, phantom, "\Downarrow\gamma"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, bend left, "S\psi"] \arrow[dr, bend right, "S\psi'"'] \arrow[dr, phantom, "\Downarrow S\delta"] & TY \arrow[r, "T\psi"] & TSZ \arrow[r, "T\alpha_Z"] & TTZ \arrow[r, "m_Z"] & TZ \\
& & SSZ \arrow[ur, bend left] \arrow[r, "\alpha_{SZ}", "S\alpha_Z"'] \arrow[dr, bend right, "m_Z"'] & STZ \arrow[ur, "\alpha_{TZ}"'] & & \\
& & & SZ \arrow[uurr, bend right, "\alpha_Z"'] & &
\end{tikzcd}$$

which follows from the naturality of $\bar{\alpha}$.

The 2-cell $\bar{h}$ is given, at each component, by

$$\begin{tikzcd}
X \arrow[rr, "u_X"] \arrow[dr, "u_X"'] & \arrow[d, phantom, "\Downarrow\bar{a}_X"] & TX \\
& SX \arrow[ur, "\alpha_X"'] &
\end{tikzcd}$$

It now remains to check coherence. We begin with the simpler task of checking the commutativity of the following two diagrams

$$\begin{tikzcd}
\mathsf{id}_{AX} * A\varphi \arrow[r, "\delta_\varphi"] \arrow[d, "\bar{h}_X \times 1"'] & A\varphi \\
A(\mathsf{id}_X) * A\varphi \arrow[r, "h"'] & A(\mathsf{id}_X * \varphi) \arrow[u, "A(\lambda_\varphi)"']
\end{tikzcd}
\qquad
\begin{tikzcd}
A\varphi * \mathsf{id}_{AY} \arrow[r, "\rho_\varphi"] \arrow[d, "1 \times \bar{h}_Y"'] & A\varphi \\
A\varphi * A(\mathsf{id}_Y) \arrow[r, "h"'] & A(\varphi * \mathsf{id}_Y) \arrow[u, "A(\rho_\varphi)"']
\end{tikzcd}
\tag{8.1}$$

where λ and ρ are part of the 2-categorical structure of $\mathsf{kl}(\tilde{S})$.

For the left-hand diagram we need to establish the equality of

$$\begin{tikzcd}
X \arrow[rr, "u_X"] \arrow[dr, "u_X"'] \arrow[ddr, bend right, "\varphi"'] & & TX \arrow[r, "T\varphi"] & TSY \arrow[r, "T\alpha_Y"] & TTY \arrow[r, "m_Y"] & TY \\
& SX \arrow[ur, "\alpha_X"'] \arrow[r, "T\varphi"'] & SSY \arrow[ur, "\alpha_{SY}"'] \arrow[r, "T\alpha_Y"'] \arrow[dr, "m_Y"'] & STY \arrow[ur, "\alpha_{TY}"'] & & \\
& SY \arrow[ur, "u_Y"'] \arrow[rr, "1_{SY}"'] & & SY \arrow[uurr, bend right, "\alpha_Y"'] & &
\end{tikzcd}
\tag{8.2}$$

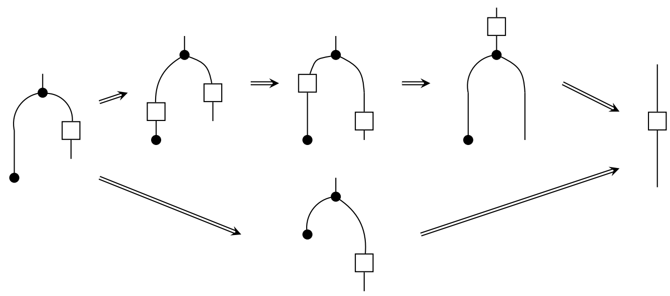
and

$$\begin{tikzcd}
X \arrow[r, "u_X"] \arrow[dr, bend right, "\varphi"'] & TX \arrow[r, "T\varphi"] & TSY \arrow[r, "T\alpha_Y"] & TTY \arrow[r, "m_Y"] & TY \\
& SY \arrow[ur, "u_{SY}"'] \arrow[r, "\alpha_Y"'] & TY \arrow[ur, "u_{TY}"'] \arrow[urr, bend right, "1_{TY}"'] & &
\end{tikzcd}$$

Note that since $\bar{h}$ is a modification, diagram 8.2 quails

$$\begin{tikzcd}
X \arrow[r, "u_X"] \arrow[dr, bend right, "\varphi"'] & TX \arrow[rr, "T\varphi"] & & TSY \arrow[r, "T\alpha_Y"] & TTY \arrow[r, "m_Y"] & TY \\
& SY \arrow[urr, "u_{SY}"] \arrow[r, "u_{SY}"'] \arrow[drr, bend right, "1_{TY}"'] & SSY \arrow[ur, "\alpha_{SY}"] \arrow[r, "S\alpha_Y"'] \arrow[dr, "m_Y"] & STY \arrow[ur, "\alpha_{TY}"] & & \\
& & & SY \arrow[uurr, bend right, "\alpha_Y"'] & &
\end{tikzcd}$$

Finally note the the diagrams are equal because the sub-diagrams that originate at SY are equal:



For the right square in 8.1 we need to show that

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, "Su_Y"] \arrow[dddrr, bend right, "1_{SY}"'] & TY \arrow[rr, "Tu_Y"] \arrow[dr, "Tu_Y"] & & TTY \arrow[r, "m_Y"] & TY \\
& & SSY \arrow[r, "\alpha_{SY}"'] \arrow[dr, "S\alpha_Y"] \arrow[ddr, bend right, "m_Y"'] & TSY \arrow[ur, "T\alpha_Y"] & & \\
& & & STY \arrow[uur, "\alpha_{TY}"'] & & \\
& & & SY \arrow[uuurr, bend right, "\alpha_Y"'] & &
\end{tikzcd}$$

is the same as

$$\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] & TY \arrow[r, "Tu_Y"] \arrow[rr, bend right, "1_{TY}"'] & TTY \arrow[r, "m_Y"] & TY
\end{tikzcd}$$

This easily follows from coherence for α (cf. diagram **??**). insert reference

It remains to show the coherence for h which amounts to showing that for

$$X \xrightarrow{\varphi} Y \xrightarrow{\psi} Z \xrightarrow{\xi} W \in \mathsf{kl}(\tilde{S})$$

the following diagram commutes

$$
\begin{tikzcd}
A\xi * (A\psi * A\varphi) \arrow[r, "\simeq"] \arrow[d, "1\times h"'] & (A\xi * A\psi) * A\varphi \arrow[d, "h\times 1"] \\
A\xi * (A(\psi * \varphi)) \arrow[d, "h"'] & A(\xi * \psi) * A\varphi \arrow[d, "h"] \\
A(\xi * (\psi * \varphi)) \arrow[r, "\simeq"'] & A((\xi * \psi) * \varphi)
\end{tikzcd}
$$

in $\mathsf{kl}(T)$.

We begin by writing down the pasting diagrams in $\mathcal{K}$ for both passages. First the upper-right passage:

$$
\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, "S\psi"'] & TY \arrow[r, "T\psi"] & TSZ \arrow[r, "T\alpha_Z"] \arrow[ddr, "TS\xi"'] & TTZ \arrow[r, "m_Z"] \arrow[dr, "T^2\xi"'] & TZ \arrow[r, "T\xi"] & TSW \arrow[r, "T\alpha_W"] & TTW \arrow[r, "m_W"] & TW \\
& & SSZ \arrow[ur, "\alpha_{SZ}"] \arrow[dr, "SS\xi"'] & & & T^2SW \arrow[ur, dashed, "m_{TW}"'] \arrow[dr, "T^2\alpha_W"] & & & \\
& & & S^3W \arrow[r, dashed, "\alpha_{S^2W}"'] \arrow[dr, "Sm_W"'] & TS^2W \arrow[r, "TS\alpha_W"'] \arrow[ur, dashed, "T\alpha_{SW}"] \arrow[dr, "Tm_W"] & TSTW \arrow[r, "T\alpha_{TW}"'] & T^3W \arrow[r, "Tm_W"'] \arrow[uur, "m_{TW}"'] & TTW \arrow[uur, "m_W"'] & \\
& & & & S^2W \arrow[r, "\alpha_{SW}"'] \arrow[ddrr, bend right, "S\alpha_W"'] \arrow[dddrr, bend right, "m_W"'] & TSW \arrow[urr, bend right=15, "T\alpha_W"'] & & & \\
& & & & & & STW \arrow[uur, "\alpha_{TW}"'] & & \\
& & & & & & SW \arrow[uuuuurr, bend right, "\alpha_W"'] & &
\end{tikzcd}
$$

(8.3)

followed by the lower-left passage

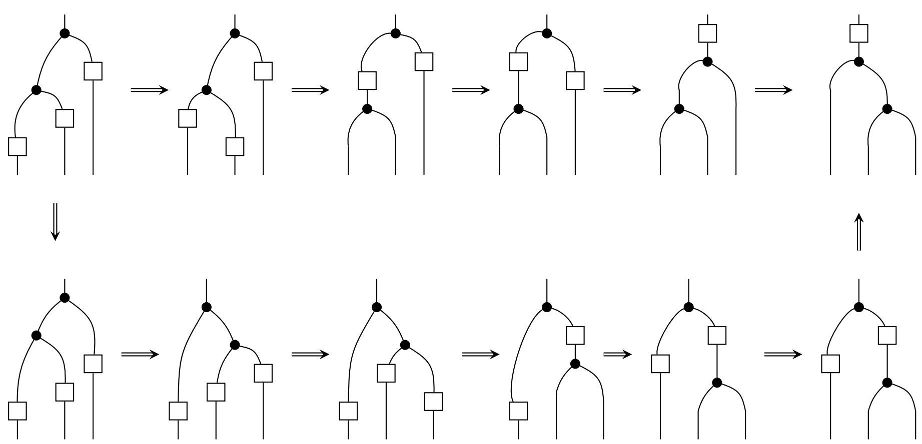
$$
\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, "S\psi"'] & TY \arrow[r, "T\psi"] & TSY \arrow[r, "T\alpha_Z"] & TTZ \arrow[r, dashed, "m_Z"] & TZ \arrow[r, dashed, "T\xi"] & TSW \arrow[r, "T\alpha_W"] & TTW \arrow[r, "m_W"] & TW \\
& & SSZ \arrow[ur, "\alpha_{SZ}"] \arrow[r, dashed, "S\alpha_Z"'] \arrow[dr, "m_Z"] \arrow[ddr, dashed, "S^2\xi"'] & STZ \arrow[ur, dashed, "\alpha_{TZ}"'] \arrow[r, phantom, "\Downarrow a"] & {} & & & & \\
& & & SZ \arrow[r, "S\xi"'] \arrow[uurr, "\alpha_Z"'] & S^2W \arrow[r, "T\alpha_Z"'] \arrow[uurr, dashed, "\alpha_{SW}"'] \arrow[dr, "m_W"'] & STW \arrow[uurr, "\alpha_{TW}"'] & & & \\
& & & S^3W \arrow[ur, dashed, "m_{TW}"'] \arrow[dr, "Tm_W"'] & & SW \arrow[uuurrr, bend right, "\alpha_W"'] & & & \\
& & & & S^2W \arrow[ur, bend right, "m_W"'] & & & &
\end{tikzcd}
$$

(8.4)

Since a is a modification, the above diagram equals

$$
\begin{tikzcd}
X \arrow[r, "\varphi"] & SY \arrow[r, "\alpha_Y"] \arrow[dr, "S\psi"'] & TY \arrow[r, "T\psi"] & TSY \arrow[r, "T\alpha_Z"] \arrow[dr, "TS\xi"] & TTZ \arrow[r, "m_Z"] \arrow[dr, "T^2\xi"] & TZ \arrow[r, "T\xi"] & TSW \arrow[r, "T\alpha_W"] & TTW \arrow[r, "m_W"] & TW \\
 & & SSZ \arrow[ur, "\alpha_{TZ}"] \arrow[ddr, bend right, "T^2\xi"'] & & TS^2W \arrow[r, dashed, "T\alpha_{SW}"'] & T^2SW \arrow[ur, dashed, "m_{TW}"'] & & & \\
 & & & & STSW \arrow[ur, "\alpha_{TSW}"] & & & & \\
 & & & S^3W \arrow[uur, dashed, "\alpha_{S^2W}"] \arrow[ur, "S\alpha_{SW}"'] \arrow[rr, "m_{SW}"'] \arrow[dr, bend right, "Tm_W"'] & & S^2W \arrow[uuur, "\alpha_{SW}"] \arrow[r, "S\alpha_W"'] \arrow[drr, bend right, "m_W"] & STW \arrow[uuur, "\alpha_{TW}"] & & \\
 & & & & & S^2W \arrow[rr, "Tm_W"'] & & SW \arrow[uuuur, bend right, "\alpha_W"'] &
\end{tikzcd}
\tag{8.5}
$$

Note that the sub-diagram to the left of the dashed line in diagram 8.3 and diagram 8.5 are equal. To complete the proof, we show that the sub-diagrams to the right of the dashed line are also equal. In the graphical proof below, the top row corresponds to the sub-diagram of 8.5 and the bottom passage corresponds to the sub-diagram of 8.3.



The equality of the two passages in the diagram follows from coherence conditions for α and the monads S and T. The details can be filled in by breaking the above diagram into regions corresponding to coherence equations for the relevant structures. □

Recalling Propositions 6.21 and 6.22, we obtain the following important corollaries.

Corollary 8.2. *Let $i : S \to T$ be a 1-cell in DL^P. Let $I : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T})$ be the 2-functor obtained by extending i to i_* (cf. Chapter 7) and then applying Theorem 8.1. Then I sends theories of type S to theories of type T sends internal distributive laws between theories of type S to internal distributive laws between theories of type T.*

8.2 Theorem 2

In this section we assume that P is the presheaf monad on Cat. Let $i : S \to T$ be a monad morphism in DL^P. Then, by Chapter 7, we can extend i to a monad morphism $i_* : \tilde{S} \to \tilde{T}$. We can also apply $(-)^*$ to every component $i_X : SX \to TX$ of i. This gives rise to an X-indexed family of profunctors $i^* : SX \to TX$. Lets assume that this family is also a monad morphism. Then via Theorem 8.1 i^* and i_* give rise to 2-functors

$$\begin{tikzcd} \mathsf{kl}(\tilde{S}) \arrow[r, bend left, "I_*"] & \mathsf{kl}(\tilde{T}) \arrow[l, bend left, "I^*"] \end{tikzcd}$$

The second theorem of this chapter shows that these functors form a lax-2-adjunction.

Theorem 8.3. *Let $\alpha : S \to T$ be a monad morphism in DL^P. Suppose that $\alpha^* : \tilde{T} \to \tilde{S}$ is also a monad morphism. Further suppose that α_* and α^* satisfy the following coherence conditions:*

- *for every $\psi : y \to x$ in* Prof*,*

$$\begin{tikzcd} Ty \arrow[r, "(\alpha_y)^*"] \arrow[dr, "1"'] & Sy \arrow[r, "\tilde{S}\psi"] \arrow[d, "(\alpha_y)_*"] & Sx \arrow[d, "(\alpha_x)_*"] \\ & Ty \arrow[r, "\tilde{T}\psi"'] & Tx \end{tikzcd} \quad = \quad \begin{tikzcd} Ty \arrow[r, "(\alpha_y)^*"] \arrow[d, "\tilde{T}\psi"'] & Sy \arrow[d, "\tilde{S}\psi"] \\ Tx \arrow[r, "(\alpha_x)^*"] \arrow[dr, "1"'] & Sx \arrow[d, "(\alpha_x)_*"] \\ & Tx \end{tikzcd} \tag{8.6}$$

- *for every category y,*

$$\begin{tikzcd} y \arrow[r, "u_y"] \arrow[dr, bend right, "u_y"'] & Sy \arrow[d, "(\alpha_y)_*"] \arrow[dr, bend left, "1_{Sy}"] & \\ & Ty \arrow[r, "(\alpha_y)^*"'] & Sy \end{tikzcd} \quad = \quad \begin{tikzcd} y \arrow[r, "u_y"] \arrow[dr, bend right, "u_y"'] & Sy \\ & Ty \arrow[u, "(\alpha_y)^*"'] \end{tikzcd} \tag{8.7}$$

- *for every category x,*

$$\begin{tikzcd} T^2Sx \arrow[r, "\tilde{T}^2(\alpha_x)_*"] \arrow[dr, "1"'] & T^3x \arrow[r, "m_{Tx}"] \arrow[d, "\tilde{T}^2(\alpha_x)^*"] & T^2x \arrow[d, "\tilde{T}(\alpha_x)^*"] \\ & T^3x \arrow[r, "m_{Sx}"'] & TSx \end{tikzcd} \quad = \quad \begin{tikzcd} T^2Sx \arrow[r, "\tilde{T}(\alpha_x)_*"] \arrow[d, "m_{Sx}"'] & T^3x \arrow[d, "m_{Tx}"] \\ TSx \arrow[r, "\tilde{T}(\alpha_x)_*"] \arrow[dr, "1"'] & T^2x \arrow[d, "\tilde{T}(\alpha_x)^*"] \\ & TSx \end{tikzcd} \tag{8.8}$$

- *for every category x,*

$$
\begin{tikzcd}
T^2x \arrow[r, "((\alpha_x)^*)^2"] \arrow[d, "m_x"'] & S^2x \arrow[d, "m_x"] \\
Tx \arrow[r, "(\alpha_x)^*"] \arrow[d, "1"'] & Sx \arrow[dl, "(\alpha_x)_*"] \\
Tx &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
T^2x \arrow[r, "((\alpha_x)^*)^2"] \arrow[dr, "1"'] & S^2x \arrow[r, "m_x"] \arrow[d, "((\alpha_x)_*)^2"] & Sx \arrow[d, "(\alpha_x)_*"] \\
& T^2x \arrow[r, "m_x"'] & Tx
\end{tikzcd}
\tag{8.9}
$$

Then the functors

$$
\begin{tikzcd}
\mathsf{kl}(\tilde{S}) \arrow[r, bend left, "A_*"] & \mathsf{kl}(\tilde{T}) \arrow[l, bend left, "A^*"]
\end{tikzcd}
$$

form a lax 2-adjunction.

Proof. Let $x \in \mathsf{kl}(\tilde{S})$ and $y \in \mathsf{kl}(\tilde{T})$. We wish to show an equivalence of categories

$$\mathsf{kl}(\tilde{S})(A^*y, x) \simeq \mathsf{kl}(\tilde{T})(y, A_*x)$$

Since A^* and A_* are both identity-on-objects, this is the same as showing the equivalence

$$\mathsf{kl}(\tilde{S})(y, x) \simeq \mathsf{kl}(\tilde{T})(y, x)$$

Define functors

$$f_{y,x}(y \overset{\psi}{\nrightarrow} Sx) = \ y \xrightarrow{\psi} Sx \xrightarrow{(\alpha_X)_*} Tx$$

and

$$g_{y,x}(\ y \xrightarrow{\psi'} Tx\ ) = \ y \xrightarrow{\psi'} Tx \xrightarrow{(\alpha_X)_*} Sx$$

By the first theorem in this section, both $f_{y,x}$ and $g_{y,x}$ are functors.

Consider

$$
\begin{aligned}
y \xrightarrow{\psi} Sx \ \to \ y \xrightarrow{\psi} Sx \xrightarrow{(\alpha_X)_*} Tx \xrightarrow{(\alpha_X)^*} SX \\
= g_{y,x}(\ y \xrightarrow{\psi} Sx \xrightarrow{(\alpha_X)_*} Tx\ ) \\
= g_{y,x} \cdot f_{y,x}(\ y \xrightarrow{\psi} Sx\ ).
\end{aligned}
$$

where the 2-cell in the first line comes from the internal adjunction $(\alpha_X)_* \dashv (\alpha_X)^*$ is Prof. Similarly, there is a 2-cell

$$f_{y,x} \cdot g_{y,x}(\ y \xrightarrow{\psi'} Tx\ ) \to (\ y \xrightarrow{\psi'} Tx\ )$$

Thus

$$f_{y,x} \dashv g_{y,x}$$

To complete the proof, we need to show that f and g are both lax-natural in x and y. We show the lax-naturality of f in y in full detail; the proofs for the other cases are similar.

For every $y' \xrightarrow{\varphi} Ty$ in $\mathsf{kl}(\tilde{T})$, we need to construct the following 2-cell in Cat

$$\begin{tikzcd}
\mathsf{kl}(\tilde{S})(y,x) \arrow[r, "\mathsf{kl}(\tilde{S})(A^*\varphi{,}x)"] \arrow[d, "f_{y,x}"'] \arrow[dr, phantom, "\Downarrow"] & \mathsf{kl}(\tilde{S})(y',x) \arrow[d, "f_{y',x}"] \\
\mathsf{kl}(\tilde{T})(y,x) \arrow[r, "\mathsf{kl}(\tilde{T})(\varphi{,}x)"'] & \mathsf{kl}(\tilde{T})(y',x)
\end{tikzcd}$$

Taking $\psi : y \nrightarrow Sx$ in $\mathsf{kl}(\tilde{S})(y,x)$, we see that the lower-left passage evaluates to

$$\mathsf{kl}(\tilde{T})(\varphi,x)\cdot f_{y,x}(\psi) = \; y' \xrightarrow{\varphi} Ty \xrightarrow{\tilde{T}\psi} TSx \xrightarrow{\tilde{T}(\alpha_x)_*} T^2x \xrightarrow{m_x} Tx$$

while the upper-right passage evaluates to

$$\begin{aligned}
f_{y',x}\cdot \mathsf{kl}(\tilde{S})(A^*\varphi,x)(\psi) &= \; y' \xrightarrow{A^*\varphi} Sy \xrightarrow{\tilde{S}\psi} S^2x \xrightarrow{m_X} Sx \xrightarrow{(\alpha_x)_*} Tx \\
&= \; y' \xrightarrow{\varphi} Ty \xrightarrow{(\alpha_y)^*} Sy \xrightarrow{\tilde{S}\psi} S^2x \xrightarrow{m_X} Sx \xrightarrow{(\alpha_x)_*} Tx
\end{aligned}$$

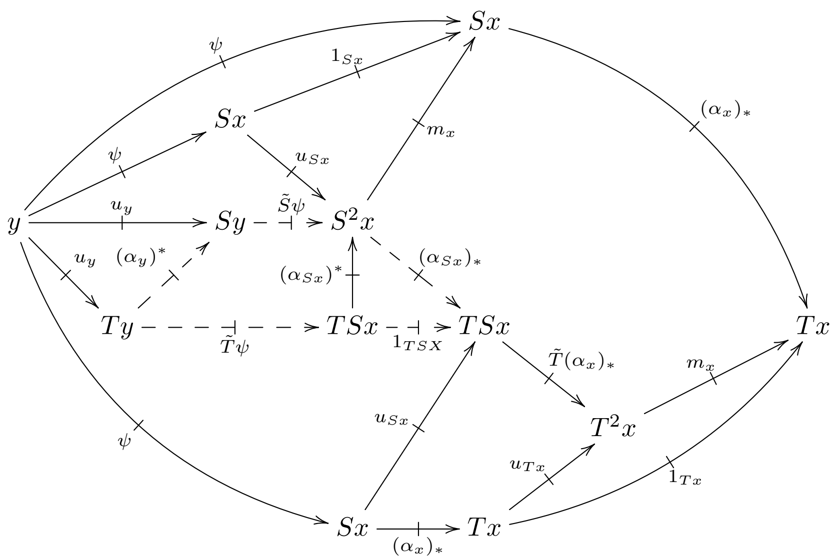
The desired 2-cell is constructed by composing the 2-cells in the following diagram

$$\begin{tikzcd}
Ty \arrow[r, "(\alpha_y)^*", "|" marking] \arrow[dr, "\tilde{T}\psi"', "|" marking] & Sy \arrow[r, "\tilde{S}\psi", "|" marking] \arrow[d, phantom, "\Downarrow(1)"] & S^2x \arrow[r, "m_x", "|" marking] \arrow[dr, "(\alpha_x)^2_*", "|" marking] \arrow[d, phantom, "\Downarrow(3)"] & Sx \arrow[r, "(\alpha_x)_*", "|" marking] \arrow[d, phantom, "\Downarrow(2)"] & Tx \\
 & TSx \arrow[ur, "(\alpha_{Sx})^*", "|" marking] \arrow[rr, "\tilde{T}(\alpha_x)_*"', "|" marking] & & T^2x \arrow[ur, "m_x"', "|" marking] &
\end{tikzcd}$$

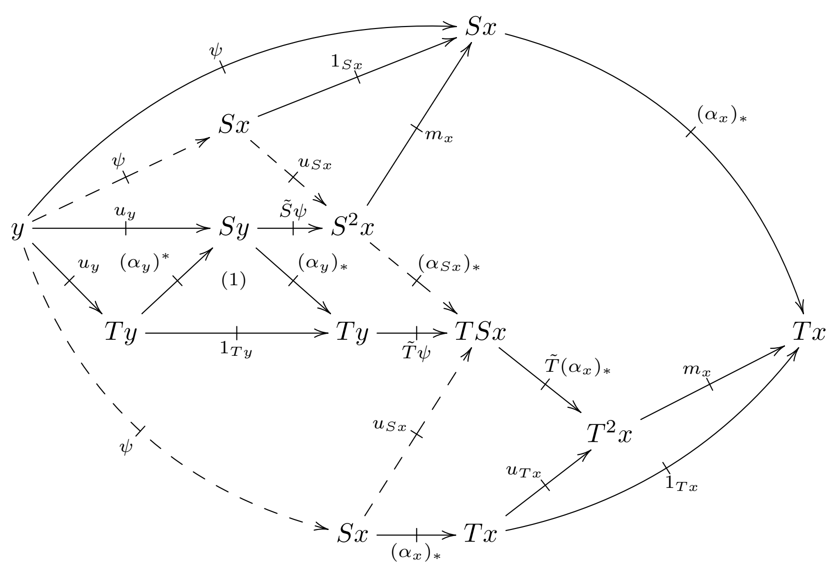
where (1) is the 2-naturality of α^*, (2) is given by the fact that α_* is a monad morphism and (3) follows from the following calculation

$$(\alpha_x)^2_*\cdot(\alpha_{Sx})^* = \;\begin{tikzcd}
TSx \arrow[r, "(\alpha_{Sx})^*", "|" marking] \arrow[rr, bend right=50, "1_{TSx}"', "|" marking] & SSx \arrow[r, "(\alpha_{Sx})_*", "|" marking] \arrow[d, phantom, "\Downarrow"] & TSx \arrow[r, "\tilde{T}(\alpha_x)_*", "|" marking] & TTx \\
 & {} & &
\end{tikzcd}$$

We check coherence conditions for this 2-cell. Diagram 4.6 of Chapter 4 translates to the requirement that the 2-cell below equals the identity (for simplicity we omit the 2-cell labels.



We apply our Equation 8.6 to the dotted part to get



(8.10)

We now focus of the diagram inside the dotted region. We show that it equals to the identity. Note that by Equation 8.7, we can remove the triangle (1). Note that the resulting diagram equals to the left-hand-side of the following equation

$$
\begin{tikzcd}
y \arrow[rr, "\psi"] \arrow[d, "u_y"] \arrow[dd, bend right=50, "u_y"'] & & Sx \arrow[d, "u_{SX}"] \arrow[dd, bend left=50, "u_{Sx}"] \arrow[dd, bend left=90, "(\alpha_{Sx})_*\cdot u_{Sx}"] \\
Sy \arrow[rr, "\tilde{S}\psi"] \arrow[d, "(\alpha_y)_*"] & & S^2x \arrow[d, "(\alpha_{Sx})_*"'] \arrow[r, phantom, "\overset{a}{\Leftarrow}"] & \arrow[r, phantom, "\overset{a^{-1}}{\Leftarrow}"] & \\
Ty \arrow[rr, "\tilde{T}\psi"'] & & TSx
\end{tikzcd}
=
\begin{tikzcd}
y \arrow[rr, "\psi"] \arrow[d, "u_y"'] & & Sx \arrow[d, "u_{Sx}"'] \arrow[d, bend left=60, "(\alpha_{Sx})_*\cdot u_{Sx}"] \arrow[d, phantom, shift left=4, "\overset{a^{-1}}{\Leftarrow}"] \\
Ty \arrow[rr, "\tilde{T}\psi"'] & & TSx
\end{tikzcd}
$$

where the equality follows since 2-natural transformations compose. Replacing the dotted region with the above equality, we obtain the left-hand-side of

$$
\begin{tikzcd}
 & Sx \arrow[r, "u_{Sx}"] \arrow[rrd, "u_{Sx}"'] & S^2x \arrow[rd, bend left, "(\alpha_{Sx})_*"] & \\
y \arrow[ru, bend left, "\psi"] \arrow[r, "u_y"] \arrow[rd, bend right, "\psi"'] & Ty \arrow[rr, "\tilde{T}\psi"] & & TSx \\
 & Sx \arrow[rru, bend right, "u_{Sx}"'] & &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
 & & S^2x \arrow[rd, bend left, "(\alpha_{Sx})_*"] & \\
y \arrow[r, "\psi"] & Sx \arrow[ru, bend left, "u_{Sx}"] \arrow[rr, "u_{Sx}"'] & & TSx
\end{tikzcd}
$$

where the quality holds since the bottom two 2-cells of the left-hand-side cancel each other out. All these reductions turn diagram 8.10 into

$$
\begin{tikzcd}
 & & & Sx \arrow[rrdd, bend left, "(\alpha_x)_*"] & & \\
y \arrow[rrru, bend left, "\psi"] \arrow[rrd, "\psi"] & & \arrow[r, phantom, "\Downarrow(3)"] & S^2x \arrow[u, "m_x"] \arrow[d, dashed, "(\alpha_{Sx})_*"] & \Downarrow(2) & \\
 & & Sx \arrow[ruu, "1_{Sx}"] \arrow[ru, dashed, "u_{Sx}"] \arrow[r, "u_{Sx}"] \arrow[rdd, dashed, "(\alpha_x)_*"'] & TSx \arrow[rd, dashed, "\tilde{T}(\alpha_x)_*"] & & Tx \\
 & & & & T^2x \arrow[ru, dashed, "m_x"] & \\
 & & & Tx \arrow[ru, "u_{Tx}"] \arrow[rruu, dashed, bend right, "1_{Tx}"'] & &
\end{tikzcd}
\tag{8.11}
$$

By coherence for 2-monad morphisms, the dotted region equals

$$
\begin{tikzcd}
 & S^2x \arrow[r, "(\alpha_{Sx})_*"] \arrow[d, "m_x"] & TSx \arrow[r, "\tilde{T}(\alpha_x)_*"] & T^2x \arrow[d, "m_x"] \\
Sx \arrow[ru, "u_{Sx}"] \arrow[r, "1_{Sx}"'] & Sx \arrow[rr, "(\alpha_x)_*"'] & & Tx
\end{tikzcd}
$$

which is the inverse of the composite 2-cell (3)+ (2). This turns diagram 8.11 into

$$
\begin{tikzcd}
y \arrow[r, "\psi"] & Sx \arrow[r, "u_{Sx}"] \arrow[rr, bend left, "1_{Sx}"] \arrow[rr, bend right, "1_{Sx}"'] & S^2 \arrow[r, "m_x"] & Sx \arrow[r, "(\alpha_x)_*"] & Tx
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
y \arrow[r, "\psi"] & Sx \arrow[r, "(\alpha_x)_*"] & Tx
\end{tikzcd}
$$

as required.

What about the third condition for coherence of 2-nat. isos?

We shift focus to diagram **??** of Chapter 4. Suppose that $\gamma : y'' \nrightarrow Ty'$ and $\beta : y' \nrightarrow Ty$ are given. Then we wish to prove the following equality

$$
\begin{tikzcd}
\mathsf{kl}(\tilde{S})(y,x) \arrow[r, "\mathsf{kl}(\tilde{S})(A^*\beta,x)"] \arrow[d, "f_{y,x}"] & \mathsf{kl}(\tilde{S})(y',x) \arrow[r, "\mathsf{kl}(\tilde{S})(A^*\gamma,x)"] \arrow[d, "f_{y',x}"] & \mathsf{kl}(\tilde{S})(y'',x) \arrow[d, "f_{y'',x}"] \\
\mathsf{kl}(\tilde{T})(y,x) \arrow[r, "\mathsf{kl}(\tilde{T})(\beta,x)"] \arrow[rr, bend right, "\mathsf{kl}(\tilde{T})(\beta\cdot\gamma,x)"'] & \mathsf{kl}(\tilde{T})(y',x) \arrow[r, "\mathsf{kl}(\tilde{T})(\gamma,x)"] & \mathsf{kl}(\tilde{T})(y'',x)
\end{tikzcd}
=
\begin{tikzcd}
 & \mathsf{kl}(\tilde{S})(y',x) \arrow[dr, bend left, "\mathsf{kl}(\tilde{S})(A^*\gamma,x)"] & \\
\mathsf{kl}(\tilde{S})(y,x) \arrow[ur, bend left, "\mathsf{kl}(\tilde{S})(A^*\beta,x)"] \arrow[rr, "\mathsf{kl}(\tilde{S})(A^*(\beta\cdot\gamma,x)"] \arrow[d, "f_{y,x}"'] & & \mathsf{kl}(\tilde{S})(y'',x) \arrow[d, "f_{y'',x}"] \\
\mathsf{kl}(\tilde{T})(y,x) \arrow[rr, "\mathsf{kl}(\tilde{T})(\beta\cdot\gamma,x)"'] & & \mathsf{kl}(\tilde{T})(y'',x)
\end{tikzcd}
\tag{8.12}
$$

Fix $\psi :\in \mathsf{kl}(\tilde{S})(y,x)$. Then the 2-cells give by the left-hand-side sends is

$$
\begin{tikzcd}
 & & Sy' \arrow[r, "\tilde{S}\beta"] & STy \arrow[r, "\tilde{S}(\alpha_y)^*"] & S^2y \arrow[r, "(\tilde{-})S^2\psi"] & S^3x \arrow[r, "\tilde{S}m_x"] & S^2x \arrow[r, "m_x"] \arrow[d, dashed, "(\alpha_x)^2_*"] & Sx \arrow[d, bend left, "(\alpha_x)_*"] \\
y'' \arrow[r, "\gamma"] & Ty' \arrow[ur, bend left, "(\alpha_y)^*"] \arrow[r, "\tilde{T}\beta"] & T^2y \arrow[ur, "(\alpha_{Ty})^*"] \arrow[r, "\tilde{T}(\alpha_y)^*"] \arrow[dr, "\tilde{T}^2\psi"'] \arrow[ddr, bend right, "m_y"'] & TSy \arrow[ur, "(\alpha_{Sy})^*"] \arrow[r, "\tilde{T}\tilde{S}\psi"] & TS^2x \arrow[ur, "(\alpha_{S^2x})^*"] \arrow[r, "\tilde{T}m_x"] \arrow[dr, dashed, "\tilde{T}(\alpha_x)^2_*"] & TSx \arrow[ur, dashed, "(\alpha_{Sx})^*"] \arrow[r, dashed, "\tilde{T}(\alpha_x)_*"] & T^2x \arrow[r, "m_x"] & Tx \\
 & & & T^2Sx \arrow[ur, dashed, "\tilde{T}(\alpha_{Sx})^*"] \arrow[rr, dashed, "\tilde{T}^2(\alpha_x)_*"'] & & T^3x \arrow[ur, "\tilde{T}m_x"'] & & \\
 & & & Ty \arrow[rr, "\tilde{T}\psi"'] & & TSx \arrow[uur, bend right, "\tilde{T}(\alpha_x)_*"'] & &
\end{tikzcd}
$$

We first expand the two dashed triangles

$$
\begin{tikzcd}
 & & Sy' \arrow[r, "\tilde{S}\beta"] & STy \arrow[r, "\tilde{S}(\alpha_y)^*"] & S^2y \arrow[r, "(\tilde{-})S^2\psi"] & S^3x \arrow[r, dashed, "\tilde{S}m_x"] & S^2x \arrow[rr, dashed, "m_x"] \arrow[d, "(\alpha_{Sx})_*"] & & Sx \arrow[dd, dashed, bend left, "(\alpha_x)_*"] \\
 & & & & & & TSx \arrow[dr, "\tilde{T}(\alpha_x)_*"] & & \\
y'' \arrow[r, "\gamma"] & Ty' \arrow[uur, bend left, "(\alpha_y)^*"] \arrow[r, "\tilde{T}\beta"] & T^2y \arrow[uur, "(\alpha_{Ty})^*"] \arrow[r, "\tilde{T}(\alpha_y)^*"] \arrow[ddr, "\tilde{T}^2\psi"'] \arrow[dddr, bend right, "m_y"'] & TSy \arrow[uur, "(\alpha_{Sy})^*"] \arrow[r, "\tilde{T}\tilde{S}\psi"] & TS^2x \arrow[uur, dashed, "(\alpha_{S^2x})^*"] \arrow[r, "\tilde{T}m_x"] \arrow[dr, "\tilde{T}(\alpha_{Sx})_*"'] & TSx \arrow[uur, "(\alpha_{Sx})^*"] \arrow[ur, "1_{TSx}"] \arrow[rr, "\tilde{T}(\alpha_x)_*"] & & T^2x \arrow[r, dashed, "m_x"] & Tx \\
 & & & & & T^2Sx \arrow[d, "\tilde{T}(\alpha_x)_*"] & & & \\
 & & & T^2Sx \arrow[uur, dashed, "\tilde{T}(\alpha_{Sx})^*"] \arrow[urr, "1_{T^2Sx}"] \arrow[rr, dashed, "\tilde{T}^2(\alpha_x)_*"'] & & T^3x \arrow[uurr, dashed, "\tilde{T}m_x"'] & & & \\
 & & & Ty \arrow[rrr, "\tilde{T}\psi"'] & & & TSx \arrow[uuurr, bend right, "\tilde{T}(\alpha_x)_*"'] & &
\end{tikzcd}
\tag{8.13}
$$

We can deduce the

$$
\begin{tikzcd}
& TSx \arrow[dl, "(\alpha_{Sx})^*"'] \arrow[d, "1"] \arrow[dr, "\tilde{T}(\alpha_x)_*"] & \\
S^2x \arrow[r, "(\alpha_{Sx})_*"'] \arrow[d, "m_x"'] & TSx \arrow[r, "\tilde{T}(\alpha_x)_*"'] & T^2x \arrow[d, "m_x"] \\
Sx \arrow[rr, "(\alpha_x)_*"'] & & Tx \\
& Tx \arrow[ul, "(\alpha_x)^*"] \arrow[ur, "1"'] &
\end{tikzcd}
=
\begin{tikzcd}
& TSx \arrow[dl, "(\alpha_{Sx})^*"'] \arrow[d, "1"] \arrow[dr, "\tilde{T}(\alpha_x)_*"] & \\
S^2x \arrow[d, "m_x"'] & TSx \arrow[l, "(\alpha_{Sx})^*"] & T^2 \arrow[l, "\tilde{T}(\alpha_x)^*"] \arrow[d, "m_x"] \\
Sx & & Tx \arrow[ll, "(\alpha_x)^*"']
\end{tikzcd}
\tag{8.14}
$$

from equation 8.9. Using this identity, we see that the dashed region in diagram 8.13 equals

$$
\begin{tikzcd}
& & & & T^2Sx \arrow[dl, bend right, "\tilde{T}(\alpha_{Sx})^*"'] \arrow[d, "1"] \arrow[dr, bend left, "\tilde{T}^2(\alpha_x)_*"] & \\
& & S^3x \arrow[ddll, bend right, "\tilde{S}m_x"'] \arrow[ddl, bend right, "m_{Sx}"] \arrow[dd, "\tilde{S}m_x"'] & TS^2x \arrow[l, "(\alpha_{S^2x})^*"'] \arrow[d, "\tilde{T}m_x"] & T^2Sx \arrow[l, "\tilde{T}(\alpha_{Sx})^*"] & T^3x \arrow[l, "\tilde{T}(\alpha_x)^*"] \arrow[dd, "\tilde{T}m_x"] \\
& & & TSx \arrow[dl, "(\alpha_{Sx})^*"'] \arrow[d, "1"] & & \\
S^2x \arrow[drr, bend right, "m_x"'] & S^2x \arrow[dr, "m_x"] & S^2x \arrow[d, "m_x"'] & TSx \arrow[l, "(\alpha_{Sx})^*"] & & T^2x \arrow[ll, "\tilde{T}(\alpha_x)^*"] \arrow[ull, "\tilde{T}(\alpha_x)^*"'] \arrow[d, "m_x"] \\
& & Sx \arrow[dr, "(\alpha_x)_*"'] & & & Tx \arrow[lll, "(\alpha_x)^*"] \\
& & & Tx \arrow[urr, "1"'] & &
\end{tikzcd}
$$

(Note that the two new 2-cells on the left are inverses to each other). Since the triangle in the middle is the identity, we can apply coherence for 2-monad morphisms to end up with

$$
\begin{tikzcd}
& & & & T^2Sx \arrow[dl, bend right, "\tilde{T}(\alpha_{Sx})^*"'] \arrow[d, "1"] \arrow[dr, bend left, "\tilde{T}^2(\alpha_x)_*"] & & \\
& S^3x \arrow[dl, bend right, "\tilde{S}m_x"'] \arrow[d, "m_{Sx}"'] & & TS^2x \arrow[ll, "(\alpha_{S^2x})^*"'] & T^2Sx \arrow[l, "\tilde{T}(\alpha_{Sx})^*"] \arrow[d, "m_{Sx}"'] & T^3x \arrow[l, "\tilde{T}(\alpha_x)^*"] \arrow[d, "m_{Tx}"] \arrow[dr, bend left, "\tilde{T}m_x"] & \\
S^2x \arrow[dr, bend right, "m_x"'] & S^2x \arrow[d, "m_x"'] & & & TSx \arrow[lll, "(\alpha_{Sx})^*"] & T^2x \arrow[l, "\tilde{T}(\alpha_x)^*"] \arrow[d, "m_x"] & T^2x \arrow[dl, bend left, "m_x"] \\
& Sx \arrow[drr, "(\alpha_x)_*"'] & & & & Tx \arrow[llll, "(\alpha_x)^*"] & \\
& & & Tx \arrow[urr, "1"'] & & &
\end{tikzcd}
$$

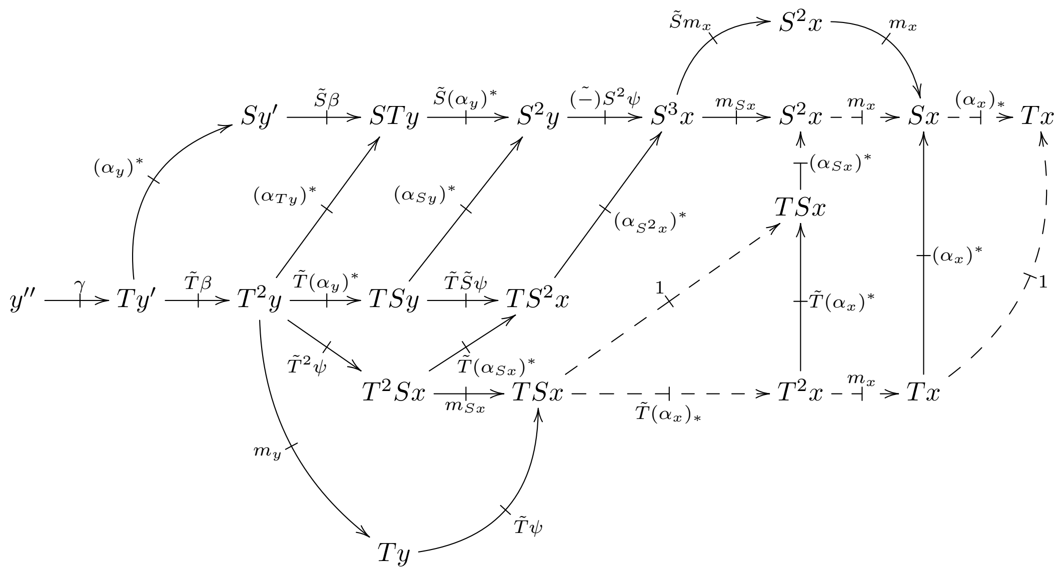
We plug this result into diagram 8.13 to obtain

$$
\begin{tikzcd}
 & & & & & & S^2x \arrow[dr, bend left, "m_x"] & & & \\
 & & Sy' \arrow[r, "\tilde S\beta"] & STy \arrow[r, "\tilde S(\alpha_y)^*"] & S^2y \arrow[r, "(\tilde{-})S^2\psi"] & S^3x \arrow[ur, bend left, "\tilde S m_x"] \arrow[r, "m_{Sx}"] & S^2x \arrow[r, "m_x"] & Sx \arrow[r, "(\alpha_x)_*"] & Tx \arrow[ddr, bend left, "1"] & \\
 & & & & & & TSx \arrow[u, "(\alpha_{Sx})^*"] & & & \\
y'' \arrow[r, "\gamma"] & Ty' \arrow[uur, bend left, "(\alpha_y)^*"] \arrow[r, "\tilde T\beta"] & T^2y \arrow[uur, "(\alpha_{Ty})^*"] \arrow[r, "\tilde T(\alpha_y)^*"] \arrow[ddr, "\tilde T^2\psi"] \arrow[dddr, bend right, "m_y"'] & TSy \arrow[uur, "(\alpha_{Sy})^*"] \arrow[r, "\tilde T\tilde S\psi"] & TS^2x \arrow[uur, "(\alpha_{S^2x})^*"] & & & T^2x \arrow[ul, dashed, "\tilde T(\alpha_x)^*"'] \arrow[rr, "m_x"] & & Tx \arrow[uull, "(\alpha_x)^*"'] \\
 & & & & & T^2Sx \arrow[ul, "\tilde T(\alpha_{Sx})^*"'] \arrow[uur, dashed, "m_{Sx}"] & & \arrow[r, phantom, "\Downarrow a"] & & \\
 & & & T^2Sx \arrow[uur, "\tilde T(\alpha_{Sx})^*"] \arrow[urr, dashed, "1_{T^2Sx}"] \arrow[rrr, dashed, "\tilde T^2(\alpha_x)_*"'] \arrow[dr, dashed] & & & T^3x \arrow[ul, "\tilde T^2(\alpha_x)^*"'] \arrow[uur, dashed, "m_{Tx}"'] \arrow[rr, "\tilde T m_x"] \arrow[dr, "m_{Tx}"] & \arrow[d, phantom, "\Downarrow a^{-1}"] & T^2x \arrow[uur, "m_x"'] & \\
 & & & Ty \arrow[r, "\tilde T\psi"'] & TSx \arrow[rrr, dashed, "\tilde T(\alpha_x)_*"'] & & & T^2x \arrow[uuurr, bend right, "m_x"'] & &
\end{tikzcd}
\tag{8.15}
$$

First, observe that the 2-cells a and a^{-1} cancel each other out. We then apply equation 8.8 to the dashed region to obtain

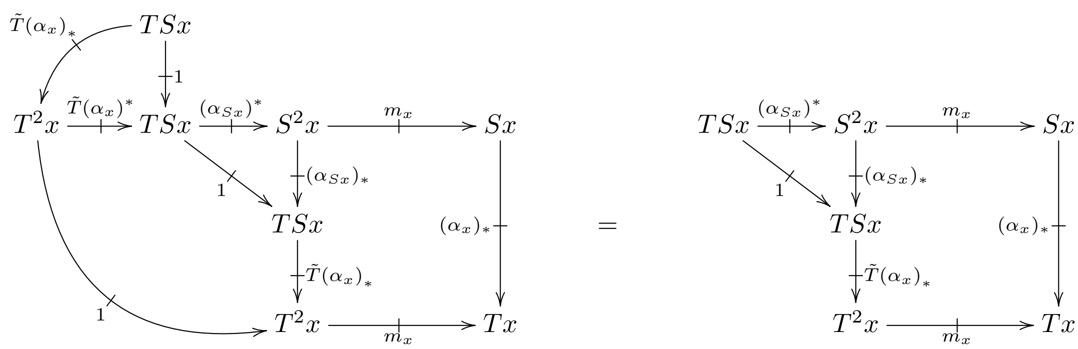
$$
\begin{tikzcd}
 & TSx \arrow[rr, "1"] \arrow[dr, "\tilde T(\alpha_x)_*"] \arrow[dd, phantom, "\Downarrow a"] & & TSx \\
T^2Sx \arrow[ur, bend left, "m_{Sx}"] \arrow[dr, "\tilde T^2(\alpha_x)_*"] \arrow[ddr, bend right, "m_{Sx}"'] & & T^2x \arrow[ur, "\tilde T(\alpha_x)^*"'] & \\
 & T^3x \arrow[ur, "m_{Tx}"] \arrow[d, phantom, "\Downarrow a^{-1}"] & & \\
 & TSx \arrow[uur, bend right, "\tilde T(\alpha_x)_*"'] & &
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
T^2Sx \arrow[r, "m_{Sx}"] & TSx \arrow[rr, "1"] \arrow[dr, "\tilde T(\alpha_x)_*"'] & & TSx \\
 & & T^2x \arrow[ur, "\tilde T(\alpha_x)^*"'] &
\end{tikzcd}
$$

This leads to

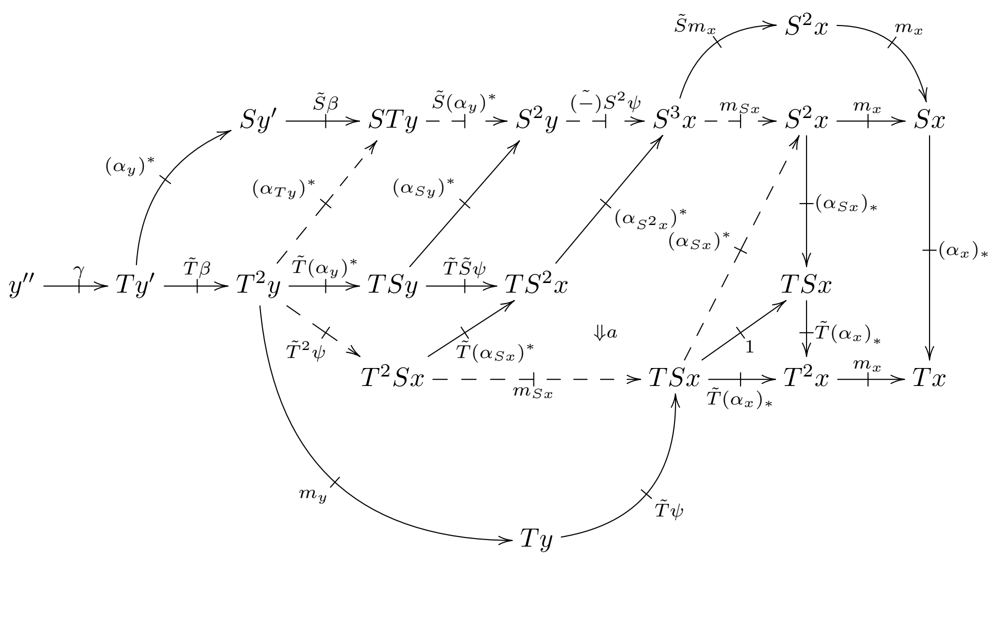


(8.16)

We apply the equation 8.9 to the dashed region transforming it to



with equality holding by one of the triangle identities. Diagram 8.16 becomes



(8.17)

Since the 2-cell a is a modification, we transform the above diagram into

$$
\begin{tikzcd}
 & & & & & & S^2x \arrow[dr, bend left, "m_x"] & \\
 & & Sy' \arrow[r, "\tilde{S}\beta"] & STy \arrow[r, "\tilde{S}(\alpha_y)^*"] & S^2y \arrow[r, "\tilde{S}^2\psi"] \arrow[d, "m_y"] & S^3x \arrow[ur, bend left, "\tilde{S}m_x"] \arrow[r, "m_{Sx}"] & S^2x \arrow[r, "m_x"] \arrow[dd, "(\alpha_{Sx})_*"] & Sx \arrow[ddd, "(\alpha_x)_*"] \\
 & & & & Sy \arrow[urr, "\tilde{S}\psi"] & & & \\
y'' \arrow[r, "\gamma"] & Ty' \arrow[r, "\tilde{T}\beta"] \arrow[uur, bend left, "(\alpha_y)^*"] & T^2y \arrow[rr, "m_y"] \arrow[uur, "(\alpha_{Ty})^*"] \arrow[dr, "\tilde{T}^2\psi"'] \arrow[ddrr, bend right, "m_y"'] & & Ty \arrow[u, "(\alpha_y)^*"] \arrow[dr, "\tilde{T}\psi"] \arrow[dl, phantom, "\Downarrow a"] & & TSx \arrow[d, "\tilde{T}(\alpha_x)_*"] & \\
 & & & T^2Sx \arrow[rr, "m_{Sx}"'] \arrow[dr, phantom, "\Downarrow a^{-1}"] & & TSx \arrow[uuur, "(\alpha_{Sx})^*"] \arrow[ur, "1"'] \arrow[r, "\tilde{T}(\alpha_x)_*"'] & T^2x \arrow[r, "m_x"] & Tx \\
 & & & & Ty \arrow[ur, bend right, "\tilde{T}\psi"'] & & &
\end{tikzcd}
\tag{8.18}
$$

We cancel a and a^{-1} and transform the diagram to get

$$
\begin{tikzcd}
 & & Sy' \arrow[r, "\tilde{S}\beta"] & STy \arrow[r, "\tilde{S}(\alpha_y)^*"] & S^2y \arrow[r, "\tilde{S}^2\psi"] \arrow[dr, "m_y"] & S^3x \arrow[r, "\tilde{S}m_x"] \arrow[dr, "m_{Sx}"] & S^2x \arrow[dr, bend left, "m_x"] & & \\
y'' \arrow[r, "\gamma"] & Ty' \arrow[r, "\tilde{T}\beta"] \arrow[ur, bend left, "(\alpha_y)^*"] & T^2y \arrow[r, "m_y"] \arrow[ur, "(\alpha_{Ty})^*"] & Ty \arrow[rr, "(\alpha_y)^*"] \arrow[drr, bend right, "\tilde{T}\psi"'] & & Sy \arrow[r, "\tilde{S}\psi"] & S^2x \arrow[r, "m_x"] \arrow[dr, "(\alpha_x)_*^2"] & Sx \arrow[dr, bend left, "(\alpha_x)_*"] & \\
 & & & & & TSx \arrow[ur, "(\alpha_{Sx})^*"] \arrow[rr, "\tilde{T}(\alpha_x)_*"'] & & T^2x \arrow[r, "m_x"] & Tx
\end{tikzcd}
\tag{8.19}
$$

which equals the right-hand-side of equation 8.12. $\square$

Chapter 9

Examples

This is the final research chapter of this thesis. It connects the general theory developed in Chapters 7 and 8 with the ideas of Categorical Universal Algebra we have explored in Chapters 2 and 6. In particular we show that monad morphisms between monads S and M and F give rise to a lax adjunction between the Kleisli bicategories of these monads. This in turn creates adjunctions betweens theories of one type and theories of another type.

Here we study two such relationships. First we show that the inclusion of monoidal categories into symmetric monoidal categories gives rise to an adjunction between (coloured) plain operads and (coloured) symmetric operads. This adjunction shows how to construct the free symmetric operad on a given plain operad.

The second example studies how the inclusion of symmetric monoidal categories into finite-product categories gives rise to an adjunction between symmetric operads and Lawvere theories.

To show these two results we have to verify many technical conditions. We begin by formally checking every condition and steadily decrease the level of detail as the chapter progresses.

9.1 Example 1: $i : M \hookrightarrow S$

We perform our analysis in stages. We check that i is a 1-cell in DL^P and then verify that it further is a monad morphism there. This allows us to apply Theorem **??** to i. Next, to apply Theorem **??** we first check that $i^* : \tilde{S} \to \tilde{M}$ is a monad morphism and that it satisfies the hypothesis of the Theorem. We finally apply Theorem **??** to the pair i^*, i_*.

9.1.1 $i : M \hookrightarrow S$ is a 1-cell in DL^P

In this section we show that, in excruciating detail, that the 2-natural transformation

$$i : M \to S, \quad i_X : MX \hookrightarrow SX$$

is a 1-cell in DL^P. We begin by defining the modification i^*

$$\begin{tikzcd}
MP \arrow[r, "\lambda"] \arrow[d, "iP"'] \arrow[dr, phantom, "\Downarrow i^*"] & PM \arrow[d, "Pi"] \\
SP \arrow[r, "\rho"'] & PS
\end{tikzcd}$$

For any category X, the top-right in passage of the diagram is given by

$$\begin{aligned}
f_1 \otimes \ldots \otimes f_n &\mapsto f_1(-) \times \ldots \times f_n(-) \\
&\mapsto \int^{x_1 \otimes \ldots \otimes x_n \in MX} f_1(x_1) \times \ldots \times f_n(x_n) \times SX(-, i_X(x_1 \otimes \ldots \otimes x_n)).
\end{aligned}$$

while the bottom-left passage equates to

$$\rho(f_1 \otimes \ldots \otimes f_n) = \int^{x_1, \ldots, x_n \in X} f_1(x_1) \times \ldots \times f_n(x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n).$$

Since taking the coend over all $x_1 \otimes \ldots \otimes x_n \in MX$ is the same as taking a coend over all $x_1, \ldots, x_n \in X$, the two passages are isomorphic. The isomorphism i^*_X is given by

$$((a_1, \ldots, a_n), t : y_1 \otimes \ldots \otimes y_n \mapsto x_1 \otimes \ldots \otimes x_n) \to ((a_1, \ldots, a_n), t). \tag{9.1}$$

It now remains to show that i^* satisfies the coherence conditions of Equation 7.1. Since the left parallelogram of that equation is a strict, we actually have to ensure that the following equality holds:

$$\begin{tikzcd}
MP^2 \arrow[r, "\lambda P"] \arrow[d, "iP^2"'] \arrow[dr, phantom, "\Downarrow i^* P"] & PMP \arrow[r, "P\lambda"] \arrow[d, "PiP"] \arrow[dr, phantom, "\Downarrow P i^*"] & P^2 M \arrow[d, "P^2 i"] \\
SP^2 \arrow[r, "\rho P"] \arrow[d, "S\mu"'] \arrow[drr, phantom, "\Downarrow \rho_\mu"] & PSP \arrow[r, "P\rho"] & P^2 S \arrow[d, "\mu S"] \\
P \arrow[rr, "\lambda"'] & & PS
\end{tikzcd}
=
\begin{tikzcd}
MP^2 \arrow[rr, "\lambda P"] \arrow[dd, "M\mu"'] & & PMP \arrow[rr, "P\lambda"] \arrow[dd, phantom, "\Downarrow \lambda_\mu"] & & P^2 M \arrow[dr, "P^2 i"] \arrow[dd, "\mu M"] & \\
& & & & & P^2 S \arrow[dd, "\mu S"] \arrow[dl, phantom, "\Downarrow \mu_i"] \\
MP \arrow[rrrr, "\lambda"] \arrow[drr, "iP"'] & & & \arrow[d, phantom, "\Downarrow i^*"] & PM \arrow[dr, "Pi"] & \\
& & SP \arrow[rrr, "\rho"'] & & & PS
\end{tikzcd} \tag{9.2}$$

We approach this task in several stages. First, we provide a detailed analysis for all five 2-cells above. Then we shall examine how these 2-cells act on the functors that make up the boundary of the diagrams.

- $i^* P$

$$\begin{tikzcd}
MP^2 \arrow[r, "\lambda P"] \arrow[d, "iP^2"'] \arrow[dr, phantom, "\Downarrow i^* P"] & PMP \arrow[d, "PiP"] \\
SP^2 \arrow[r, "\rho P"'] & PSP
\end{tikzcd}$$

The 2-cell is a natural transformation that evaluates to

$$\int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1)\times\ldots\times F_n(f_n)\times SPX(-,iP(f_1\otimes\ldots\otimes f_n))$$
$$\to\int^{f_1,\ldots,f_n\in PX} F_1(f_1)\times\ldots\times F_n(f_n)\times SPX(-,f_1\otimes\ldots\otimes f_n).$$

- Pi^*

We compute

$$Pi^*: P(Pi\cdot\lambda)\to P(\rho\cdot iP)$$

as follows: for any $G\in PMPX$, we have

$$\begin{aligned}P(Pi\cdot\lambda)(G) &= \int^{f_1\otimes\ldots\otimes f_n\in MPX} G(f_1\otimes\ldots\otimes f_n)\times PSX(-,Pi\cdot\lambda(f_1\otimes\ldots\otimes f_n))\\ &\to\int^{f_1\otimes\ldots\otimes f_n\in MPX} G(f_1\otimes\ldots\otimes f_n)\times PSX(-,\rho\cdot iP(f_1\otimes\ldots\otimes f_n))\\ &= P(\rho\cdot iP)(G)\end{aligned}$$

Thus Pi^* sends

$$(g\in G(f_1\otimes\ldots\otimes f_n),\gamma\in PSX(K,Pi\cdot\lambda(f_1\otimes\ldots\otimes f_n))\mapsto(g,i^*\cdot\gamma)$$

- λ_μ

The following Lemma will greatly help us.

Lemma 9.1. *For any presheaf $h\in PX$, we have the following isomorphism*

$$\int^{f\in PX} PX(f,h)\times f(x)\simeq h(x)$$

Proof. The proof follows easily from the Yoneda Lemma:

$$\int^{f\in PX} PX(f,h)\times f(x)\simeq\int^{f\in PX} PX(f,h)\times PX(y(x),f)\simeq h(x).$$

For our purposes it is important to note that any tuple $(\gamma: f\to h,a)\in\int^{f\in PX} PX(f,h)\times f(x)$, is always equivalent to $(\mathrm{id}_h,\gamma_x(a))$ and so the above isomorphism sends (γ,a) to $\gamma_x(a)\in h(x)$. □

Given a category X, we have

$$\begin{aligned}&\mu_{MX}\cdot P\lambda_X\cdot\lambda_{PX}(F_1\otimes\ldots\otimes F_n)=\mu_{MX}(P\lambda_X\big(F_1(-)\times\ldots\times F_n(-)\big))\\ &=\mu_{MX}\big(\int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1)\times\ldots\times F_n(f_n)\times PMX(-,f_1(-)\times\ldots\times f_n(-))\big)\\ &=\int^{G\in PMX}\int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1)\times\ldots\times F_n(f_n)\times PMX(G,f_1(-)\times\ldots\times f_n(-))\times G(-).\end{aligned}$$

An application of the Lemma shows that the above presheaf is isomorphic to

$$\int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1)\times\ldots\times F_n(f_n)\times f_1(-)\times\ldots\times f_n(-)$$

On the other hand, the bottom-right passage in the diagram evaluates to

$$\begin{aligned}
&\lambda_X\cdot M\mu_X(F_1\otimes\ldots\otimes F_n)=\lambda_X(\mu_X(F_1)\otimes\ldots\otimes\mu_X(F_n))\\
&\quad=\mu_X(F_1)(-)\times\ldots\times\mu_X(F_n)(-)\\
&\quad=\int^{f_1\in PX} F_1(f_1)\times f_1(-)\times\ldots\times\int^{f_n\in PX} F_n(f_n)\times f_n(-)
\end{aligned}$$

Now pick a tuple $x_1,\ldots,x_n\in X$. A typical element in $\mu_{MX}\cdot P\lambda_X\cdot\lambda_{PX}(F_1\otimes\ldots\otimes F_n)(x_1\otimes\ldots\otimes x_n)$ is of the form

$$T=((a_1,\ldots,a_n),\gamma,y),\quad \gamma:G\to f_1(-)\times\ldots\times f_n(-),\;\; y\in G(x_1\otimes\ldots\otimes x_n).$$

and suppose that

$$\gamma_{x_1\otimes\ldots\otimes x_n}(y)=(z_1,\ldots,z_n),\text{ with each } z_i\in f_i(x_i).$$

then the natural transformation $(\lambda_\mu)_X$ sends T to the element

$$(a_1,z_1,\ldots,a_n,z_n)\in\lambda_X\cdot M\mu_X(F_1\otimes\ldots\otimes F_n)(x_1\otimes\ldots\otimes x_n)$$

- ρ_μ

Since the calculation of ρ_μ is similar to that of λ_μ, we omit some details.

$$\begin{aligned}
&\mu_{SX}\cdot P\rho\cdot\rho P(F_1\otimes\ldots\otimes F_n)\\
&\quad=\int^{G\in PSX}\int^{f_1\otimes\ldots\otimes f_n\in SPX}\rho(F_1\otimes\ldots\otimes F_n)(f_1\otimes\ldots\otimes f_n)\times PSX(G,\rho(f_1\otimes\ldots\otimes f_n))\times G(-)\\
&\quad\simeq\int^{f_1\otimes\ldots\otimes f_n\in SPX}\rho(F_1\otimes\ldots\otimes F_n)(f_1\otimes\ldots\otimes f_n)\times\rho(f_1\otimes\ldots\otimes f_n)(-)\\
&\quad=\int^{f_1\otimes\ldots\otimes f_n\in SPX}\Big(\int^{g_1,\ldots,g_n\in PX}F_1(g_1)\times\ldots\times F_n(g_n)\times SPX(f_1\otimes\ldots\otimes f_n,g_1\otimes\ldots\otimes g_n)\Big)\\
&\qquad\times\rho(f_1\otimes\ldots\otimes f_n)(-)\\
&\quad\simeq\int^{g_1,\ldots,g_n\in PX}F_1(g_1)\times\ldots\times F_n(g_n)\times\rho(g_1\otimes\ldots\otimes g_n)(-)\\
&\quad=\int^{g_1,\ldots,g_n\in PX}F_1(g_1)\times\ldots\times F_n(g_n)\times\int^{x_1,\ldots,x_n\in X}g_1(x_1)\times\ldots\times g_n(x_n)\times SX(-,x_1\otimes\ldots\otimes x_n).
\end{aligned}$$

The last isomorphisms in the above calculation is derive from the fact that every presheaf is a colimit of representables. Explicitly, the isomorphism

$$\int^{f_1\otimes\ldots\otimes f_n\in SPX} SPX(f_1\otimes\ldots\otimes f_n,g_1\otimes\ldots\otimes g_n)\times\rho(f_1\otimes\ldots\otimes f_n)(y_1\otimes\ldots\otimes y_n)\simeq\rho(g_1\otimes\ldots\otimes g_n)(y_1\otimes\ldots\otimes y_n)$$

sends the tuple $(\sigma, (a_1, \ldots, a_n))$ to $\rho(\sigma)_{y_1 \otimes \ldots \otimes y_n}(a_1, \ldots, a_n)$. Diving deeper, we observe that σ can be represented as

$$\sigma = t_1 \otimes \ldots \otimes t_m$$

where each t_i is either a natural transformation $t_i : f_k \to g_k$ or it permutes f_k and f_{k+1}. Let $(a_1, \ldots, a_n) \in f_1(y_1) \times \ldots \times f_n(y_n)$. Then

$$\rho(\sigma)_{y_1 \otimes \ldots \otimes y_n}(a_1, \ldots, a_n) = (b_1, \ldots, b_n)$$

where $b_k = (t_i)_{y_1, \ldots, y_n}(a)$ if t_i a natural transformation or $b_k = a_{k-1}$ and $b_{k-1} = a_k$, if t_i is a permutation.

The bottom-right passage for the 2-cell ρ_μ evaluates to

$$\begin{aligned}
&\rho \cdot S\mu_X(F_1 \otimes \ldots \otimes F_n) = \rho(\mu_X(F_1)(-) \times \ldots \times \mu_X(F_n)(-)) \\
&= \int^{x_1, \ldots, x_n \in X} \mu_X(F_1)(x_1) \times \ldots \times \mu_X(F_n)(x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\
&= \int^{x_1, \ldots, x_n \in X} \int^{g_1 \in PX} F_1(g_1) \times g_1(x_1) \times \ldots \times \int^{g_n \in PX} F_n(g_n) \times g_n(x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n).
\end{aligned}$$

A typical element in $\mu_{SX} \cdot P\rho \cdot \rho P(F_1 \otimes \ldots \otimes F_n)(y_1 \otimes \ldots \otimes y_n)$ is of the form

$$((a_1, \ldots, a_n), \sigma, \gamma, y)$$

where $a_i \in F_i(g_i)$, $\sigma \in SPX(f_1 \otimes \ldots \otimes f_n, g_1 \otimes \ldots \otimes g_n)$, γ is a natural transformation from G to $\rho(f_1 \otimes \ldots \otimes f_n)$ where we write

$$\gamma_{y_1 \otimes \ldots \otimes y_n}(y) = (z_1, \ldots, z_n, t) \in f_1(x_1) \times \ldots \times f_n(x_n) \times SX(y_1 \otimes \ldots \otimes y_n, x_1 \otimes \ldots \otimes x_n)$$

and $y \in G(y_1 \otimes \ldots \otimes y_n)$.

The first isomorphism sends this element to

$$\big((a_1, \ldots, a_n), \sigma, (z_1, \ldots, z_n, t)\big);$$

the second isomorphisms sends it to

$$\big((a_1, \ldots, a_n), \rho(\sigma)_{y_1 \otimes \ldots \otimes y_n}(z_1, \ldots, z_n), t\big) = \big((a_1, \ldots, a_n), (w_1, \ldots, w_n), t\big)$$

and finally, the in the last step we send this b-tuple to

$$\big((a_1, w_1), \ldots, (a_n, w_n), t\big)$$

- μ_i

Let $G : PMX^{\mathrm{op}} \to \mathrm{Set}$. Then the 2-cell is given by the passage

$$\begin{aligned}
&\mu_{SX} \cdot P^2 i_X(G) = \mu_{SX}\big(\int^{g\in PMX} G(g) \times PSX(-, Pi(g))\big) \\
&\quad = \mu_{SX}\big(\int^{g\in PMX} G(g) \times PSX(-, \int^{x_1\otimes\ldots\otimes x_n \in MX} g(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\big) \\
&\quad = \int^{H\in PSX} \Big(\int^{g\in PMX} G(g) \times PSX(H, \int^{x_1\otimes\ldots\otimes x_n\in MX} g(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\Big) \times H(-) \\
&\quad \to \int^{g\in PMX} G(g) \times \int^{x_1\otimes\ldots\otimes x_n\in MX} g(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\
&\quad \simeq \int^{x_1\otimes\ldots\otimes x_n\ \in MX} \int^{g\in PMX} G(g) \times g(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\
&\quad = Pi_X \cdot \mu_{MX}(G).
\end{aligned}$$

Fix $y_1 \otimes \ldots \otimes y_n \in MX$, $H \in PSX$ and $g \in PMX$. The a typical element in $\mu_{SX} \cdot P^2 i_X(G)(y_1 \otimes \ldots \otimes y_n)$ is (x, γ, y) with $x \in G(g)$, $\gamma : H \to \int^{x_1\otimes\ldots\otimes x_n\in MX} g(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n)$ a natural transformation and $y \in H(y_1 \otimes \ldots \otimes y_n)$. The passage sends

$$(x, \gamma, y) \mapsto (x, \gamma_{y_1\otimes\ldots\otimes y_n}(y)).$$

Showing the equality

Now that we fully understand how each 2-cell in Equation 9.2, we can finally show the desired equality.

We begin with the left-hand Diagram of 9.2. The composite 2-cell evaluates to to

$$\begin{aligned}
&\mu_{SX} \cdot P^2 i_X \cdot P\lambda_X \cdot \lambda_{PX}(F_1 \otimes \ldots \otimes F_n) \\
&\quad = \int^{H\in PSX} \Big(\int^{G\in PMX} P\lambda_X \cdot \lambda_{PX}(F_1 \otimes \ldots \otimes F_n)(G) \\
&\qquad\qquad \times PSX(H, \int^{x_1\otimes\ldots\otimes x_n\in MX} G(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\Big) \times H(-) \\
&\quad = \int^{H\in PSX} \Big(\int^{G\in PMX} \int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \times PMX(G, f_1(-) \times \ldots \times f_n(-)) \\
&\qquad\qquad \times PSX(H, \int^{x_1\otimes\ldots\otimes x_n\in MX} G(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\Big) \times H(-) \\
&\quad \to \int^{x_1\otimes\ldots\otimes x_n\in MX} \int^{G\in PMX} \int^{f_1\otimes\ldots\otimes f_n\in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \times \\
&\qquad\qquad PMX(G, f_1(-) \times \ldots \times f_n(-)) \times G(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\
&\quad \to \int^{x_1\otimes\ldots\otimes x_n\in MX} \Big(\int^{f_1\in PX} F_1(f_1) \times f_1(-) \times \ldots \times \int^{f_n\in PX} F_n(f_n) \times f_n(-)\Big) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\
&\quad \to \int^{x_1,\ldots,x_n\in X} \Big(\int^{f_1\in PX} F_1(f_1) \times f_1(-) \times \ldots \times \int^{f_n\in PX} F_n(f_n) \times f_n(-)\Big) \times SX(-, x_1 \otimes \ldots \otimes x_n)
\end{aligned}$$

Concretely, we stat by fixing H, G and $f_1 \otimes \ldots \otimes f_n$. Then a typical element of $\mu_{SX} \cdot P^2 i_X \cdot P\lambda_X \cdot \lambda_{PX}(F_1 \otimes \ldots \otimes F_n)(y_1 \otimes \ldots \otimes y_n)$ is the n-tuple $\big((a_1, \ldots, a_n), \gamma, \Gamma, y\big)$ where

- $(a_1, \ldots, a_n) \in F_1(f_1) \times \ldots \times F_n(f_n)$;
- $y \in H(y_1 \otimes \ldots \otimes y_n)$;
- $\Gamma : H \to \int^{x_1 \otimes \ldots \otimes x_n \in MX} G(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n)$; we write

  $$\Gamma_{y_1 \otimes \ldots \otimes y_n}(y) = (x \in G(x_1 \otimes \ldots \otimes x_n), t : y_1 \otimes \ldots \otimes y_n \to x_1 \otimes \ldots \otimes x_n).$$

  for some $x_1 \otimes \ldots \otimes x_n$.
- $\gamma : G \to f_1(-) \times \ldots \times f_n(-)$ with

  $$\gamma_{x_1 \otimes \ldots \otimes x_n}(x) = (z_1, \ldots, z_n) \in f_1(x_1) \times \ldots \times f_n(x_n)$$

The 2-cell acts as follows:

$$\begin{aligned}
\big((a_1, \ldots, a_n), \gamma, \Gamma, y\big) &\overset{\mu_i}{\longmapsto} \big((a_1, \ldots, a_n), \gamma, x, t\big) \\
&\overset{\lambda_\mu}{\longmapsto} \big((a_1, z_1), \ldots, (a_n, z_n), t\big) \\
&\overset{i^*}{\longmapsto} \big((a_1, z_1), \ldots, (a_n, z_n), t\big).
\end{aligned}$$

Next, lets examine what happen in the left Diagram of 9.2.

$$
\begin{aligned}
&\mu_{SX} \cdot P^2 i_X \cdot P\lambda_X \cdot \lambda_{PX}(F_1 \otimes \ldots \otimes F_n) \\
&\quad = \mu_{SX}\Big(\int^{G \in PMX} \int^{f_1 \otimes \ldots \otimes f_n \in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \times PMX(G, f_1(-) \times \ldots \times f_n(-)) \\
&\qquad\quad \times PSX(H, \int^{x_1 \otimes \ldots \otimes x_n \in MX} G(x_1 \otimes \ldots \otimes x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\Big) \\
&\quad \xrightarrow{r} \mu_{SX}\Big(\int^{f_1 \otimes \ldots \otimes f_n \in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \\
&\qquad\quad \times PSX(H, \int^{x_1 \otimes \ldots \otimes x_n \in MX} f_1(x_1) \times \ldots \times f_n(x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n))\Big) \\
&\quad = \mu_{SX}\Big(\int^{f_1 \otimes \ldots \otimes f_n \in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \times PSX\big(-, Pi_X \cdot \lambda_X(f_1 \otimes \ldots \otimes f_n)\big)\Big) \\
&\quad \to \mu_{SX}\Big(\int^{f_1 \otimes \ldots \otimes f_n \in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \times PSX\big(-, \rho(f_1 \otimes \ldots \otimes f_n)\big)\Big) \\
&\quad \xrightarrow{l} \mu_{SX}\Big(\int^{g_1 \otimes \ldots \otimes g_n \in SPX} \int^{f_1 \otimes \ldots \otimes f_n \in MPX} F_1(f_1) \times \ldots \times F_n(f_n) \\
&\qquad\quad \times SPX(g_1 \otimes \ldots \otimes g_n, f_1 \otimes \ldots \otimes f_n) \times PSX\big(-, \rho(g_1 \otimes \ldots \otimes g_n)\big)\Big) \\
&\quad \to \mu_{SX}\Big(\int^{g_1 \otimes \ldots \otimes g_n \in SPX} \int^{f_1, \ldots, f_n \in PX} F_1(f_1) \times \ldots \times F_n(f_n) \\
&\qquad\quad \times SPX(g_1 \otimes \ldots \otimes g_n f_1 \otimes \ldots \otimes f_n) \times PSX(-, \rho(g_1 \otimes \ldots \otimes g_n))\Big) \\
&\quad = \int^{G \in PSX} \Big(\int^{g_1 \otimes \ldots \otimes g_n \in SPX} \int^{f_1, \ldots, f_n \in PX} F_1(f_1) \times \ldots \times F_n(f_n) \\
&\qquad\quad \times SPX(g_1 \otimes \ldots \otimes g_n f_1 \otimes \ldots \otimes f_n) \times PSX(G, \rho(g_1 \otimes \ldots \otimes g_n))\Big) \times G(-) \\
&\quad \to \int^{x_1, \ldots, x_n \in X} \int^{g_1 \in PX} F_1(g_1) \times g_1(x_1) \times \ldots \times \int^{g_n \in PX} F_n(g_n) \times g_n(x_n) \times SX(-, \rho(x_1 \otimes \ldots \otimes x_n)).
\end{aligned}
$$

($r : P^2 i \cdot P\lambda \to P(Pi \cdot \lambda)$ and $l : P(\rho \cdot iP) \to P\rho \cdot PiP$ are the standard isomorphisms).

This time the mapping acts as follows:

$$
\begin{aligned}
\big((a_1, \ldots, a_n), \gamma, \Gamma, y\big) &\xmapsto{r} \big((a_1, \ldots, a_n), \gamma', y\big) \\
&\xmapsto{Pi^*} \big((a_1, \ldots, a_n), (i^* \cdot \gamma'), y\big) \\
&\xmapsto{l} \big((a_1, \ldots, a_n), \mathsf{id}_{f_1 \otimes \ldots \otimes f_n}, (i^* \cdot \gamma'), y\big) \\
&\xmapsto{i^* P} \big((a_1, \ldots, a_n), \mathsf{id}_{f_1 \otimes \ldots \otimes f_n}, (i^* \cdot \gamma'), y\big) \\
&\xmapsto{\rho_\mu} \big((a_1, z_1), \ldots, (a_n, z_n), t\big)
\end{aligned}
$$

where $\gamma'_{y_1 \otimes \ldots \otimes y_n} = \big(\gamma_{x_1 \otimes \ldots \otimes x_n} \cdot \pi_1(\Gamma_{y_1 \otimes \ldots \otimes y_n}), \pi_2(\Gamma_{y_1 \otimes \ldots \otimes y_n})\big)$. In particular,

$$
\gamma'_{y_1 \otimes \ldots \otimes y_n}(y) = \big((z_1, \ldots, z_n), t\big)
$$

Thus we have proved that the two composite 2-cells in Diagram 9.2 are indeed equal.

It remains to show the following equality

$$
\begin{tikzcd}
& MPX \arrow[dr, "\lambda_X"] \arrow[d, phantom, "\Downarrow(1)"] & \\
MX \arrow[ur, "M\eta_X"] \arrow[rr, "\eta_{MX}"] \arrow[dd, "i_X"'] \arrow[ddrr, phantom, "\Downarrow(2)"] & & PMX \arrow[dd, "Pi_X"] \\
& & \\
SX \arrow[rr, "\eta_{SX}"'] & & PSX
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
& MPX \arrow[dr, "\lambda_X"] \arrow[dd, "i_{PX}"] & \\
MX \arrow[ur, "M\eta_X"] \arrow[dd, "i_X"'] \arrow[r, phantom, "\Downarrow(3)"] & & PMX \arrow[dd, "Pi_X"] \arrow[l, phantom, "\Downarrow(4)"] \\
& SPX \arrow[dr, "\rho_X"] \arrow[d, phantom, "\Downarrow(5)"] & \\
DX \arrow[ur, "S\eta_X"] \arrow[rr, "\eta_{SX}"'] & & PSX
\end{tikzcd}
$$

which we do in less detail than above.

The left-hand-side gives

$$
\begin{aligned}
Pi_X \cdot \lambda_X \cdot M\eta_X(x_1 \otimes \ldots \otimes x_n) &= \int^{y_1 \otimes \ldots \otimes y_n \in MX} X(y_1, x_1) \times \ldots \times X(y_n, x_n) \times SX(-, y_1 \otimes \ldots \otimes y_n) \\
&\overset{1}{\to} \int^{y_1 \otimes \ldots \otimes y_n \in MX} MX(y_1 \otimes \ldots \otimes y_n, x_1 \otimes \ldots \otimes x_n) \times SX(-, y_1 \otimes \ldots \otimes y_n) \\
&= SX(-, x_1 \otimes \ldots \otimes x_n) \\
&\overset{2}{=} \eta_{SX} \cdot i_X(x_1 \otimes \ldots \otimes x_n).
\end{aligned}
$$

Let $(f_1, \ldots, f_n, \sigma) \in X(y_1, x_1) \times \ldots \times X(y_n, x_n) \times SX(z_1 \otimes \ldots \otimes z_n, y_1 \otimes \ldots \otimes y_n)$. Then the above passage sends it to

$$
(f_1, \ldots, f_n, \sigma) \mapsto (f_1 \otimes \ldots \otimes f_n, \sigma) \mapsto \sigma \cdot f_1 \otimes \ldots \otimes f_n
$$

Right-hand-side computes to

$$
\begin{aligned}
Pi_X \cdot \lambda_X \cdot M\eta_X(x_1 \otimes \ldots \otimes x_n) &= \int^{y_1 \otimes \ldots \otimes y_n \in MX} X(y_1, x_1) \times \ldots \times X(y_n, x_n) \times SX(-, y_1 \otimes \ldots \otimes y_n) \\
&\overset{4,3}{\to} \int^{y_1, \ldots, y_n \in X} X(y_1, x_1) \times \ldots \times X(y_n, x_n) \times SX(-, y_1 \otimes \ldots \otimes y_n) \\
&\to SX(-, x_1 \otimes \ldots \otimes x_n);
\end{aligned}
$$

it sends $(f_1, \ldots, f_n, \sigma)$ to $\sigma \cdot f_1 \otimes \ldots \otimes f_n$.

Thus the two 2-cells are equals and our proof is complete.

9.1.2 $\tilde{i_X} : \tilde{M}X \hookrightarrow \tilde{S}X$ is a monad morphism.

$i : S \to M$ is a monad morphism. Since both λ and ρ are 2-distributive laws of monads over monads, the triples $((M, \lambda), m, u)$ and $((S, \rho), m, u)$ are monads in DL^P. It turns out that (i, i^*) is a monad morphism between these monads. To prove this, we need to show that the identity 2-cell

$$
\begin{tikzcd}
M^2X \arrow[r, "i^2_X"] \arrow[d, "m_X"'] & S^2X \arrow[d, "m_X"] \\
MX \arrow[r, "i_X"'] & SX
\end{tikzcd}
$$

is a 2-cell in DL^P. This amounts to proving that the following two 2-cells are equal

$$
\begin{tikzcd}
 & M^2PX \arrow[r, "\lambda^2_X"] \arrow[dl, "i^2_{PX}"'] \arrow[d, "m_{PX}"] & PM^2X \arrow[d, "Pm_X"] \\
S^2PX \arrow[dr, "m_{PX}"'] & MPX \arrow[r, "\lambda_X"] \arrow[d, "i_{PX}"] & PMX \arrow[d, "Pi_X"] \\
 & SPX \arrow[r, "\rho_X"'] & PSX
\end{tikzcd}
=
\begin{tikzcd}
M^2PX \arrow[r, "\lambda^2_X"] \arrow[d, "iP^2_X"'] & PM^2X \arrow[d, "Pi^2_X"] \arrow[dr, "Pm_X"] & \\
S^2PX \arrow[r, "\rho^2_X"] \arrow[d, "m_{PX}"'] & PS^2X \arrow[d, "Pm_X"] & PMX \arrow[dl, "Pi_X"] \\
SPX \arrow[r, "\rho_X"'] & PSX &
\end{tikzcd}
\tag{9.3}
$$

Instead of proving this equality directly, we shall simplify it significantly by exploiting the following Proposition.

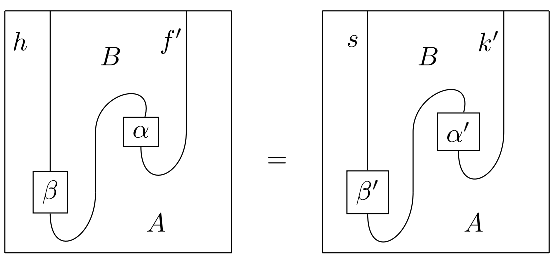
Proposition 9.2. *Let $\mathcal{K}$ be a strict 2-category. Write f' for the right adjoint of the 1-cell f. Then, provided that all the right adjoints below exist, we have*

$$
\begin{tikzcd}
A \arrow[r, bend left=60, "f"{name=F}] \arrow[r, "g"'{name=G}] \arrow[r, bend right=60, "h"'{name=H}] \arrow[r, phantom, shift left=3, "\Downarrow\alpha"] \arrow[r, phantom, shift right=4, "\Downarrow\beta"] & B
\end{tikzcd}
=
\begin{tikzcd}
A \arrow[r, bend left=60, "k"] \arrow[r, "t"'] \arrow[r, bend right=60, "s"'] \arrow[r, phantom, shift left=3, "\Downarrow\alpha'"] \arrow[r, phantom, shift right=4, "\Downarrow\beta'"] & B
\end{tikzcd}
$$

if, and only if,

$$
\begin{tikzcd}
A \arrow[r, bend left=60, "1_A"] \arrow[r, "g'\cdot h"'] \arrow[r, bend right=60, "f'\cdot h"'] \arrow[r, phantom, shift left=3, "\Downarrow b"] \arrow[r, phantom, shift right=4, "\Downarrow a"] & A
\end{tikzcd}
=
\begin{tikzcd}
A \arrow[r, bend left=60, "1_A"] \arrow[r, "t'\cdot s"'] \arrow[r, bend right=60, "k'\cdot s"'] \arrow[r, phantom, shift left=3, "\Downarrow b'"] \arrow[r, phantom, shift right=4, "\Downarrow a'"] & A
\end{tikzcd}
\tag{9.4}
$$

Proof. The proof follows easily from the observation that the string diagram for Equation 9.4 is



□

We want apply the proposition to Diagram 9.3. To do that, we first note that for any functor $f : X \to Y$,

Pf always has a right adjoint κ_f given by

$$\kappa_f : PY \to PX, \quad \varphi \mapsto \varphi \cdot f^{\mathrm{op}}.$$

Thus to prove the equality in 9.3 it suffices to show the equality of the outer diagrams in

$$\begin{tikzcd}
M^2PX \arrow[r, "\lambda^2_X"] \arrow[d, "m_{PX}"'] & PM^2X \arrow[d, bend right, "Pm_X"'] \\
MPX \arrow[r, "\lambda_X"] \arrow[d, "i_{PX}"'] & PMX \arrow[u, bend right, "\kappa_{m_X}"'] \arrow[d, bend right, "Pi_X"'] \\
SPX \arrow[r, "\rho_X"'] & PSX \arrow[u, bend right, "\kappa_{i_X}"']
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
M^2PX \arrow[r, "\lambda^2_X"] \arrow[d, "iP^2_X"'] & PM^2X \arrow[d, bend right, "Pi^2_X"'] \arrow[dr, "Pm_X"] & \\
S^2PX \arrow[r, "\rho^2_X"] \arrow[d, "m_{PX}"'] & PS^2X \arrow[u, bend right, "\kappa_{i^2_X}"'] \arrow[d, bend right, "Pm_X"'] & PMX \arrow[ul, bend right, "\kappa_{m_X}"'] \arrow[dl, "Pi_X"'] \\
SPX \arrow[r, "\rho_X"'] & PSX \arrow[u, bend right, "\kappa_{m_X}"'] \arrow[ur, bend right, "\kappa_{i_X}"'] &
\end{tikzcd}$$

Further, note that the left-hand-side of both diagrams is equal and

$$\kappa_{m_X} \cdot \kappa_{i_X} = \kappa_{m_X} \cdot \kappa_{i^2_X}.$$

Hence we need to exhibit the much simpler equality:

$$\begin{tikzcd}
M^2PX \arrow[r, "\lambda^2_X"] \arrow[d, "m_{PX}"'] \arrow[dr, phantom, "\Downarrow(1)"] & PM^2X \\
MPX \arrow[r, "\lambda_X"] \arrow[d, "i_{PX}"'] \arrow[dr, phantom, "\Downarrow(2)"] & PMX \arrow[u, "\kappa_{m_X}"'] \\
SPX \arrow[r, "\rho_X"'] & PSX \arrow[u, "\kappa_{i_X}"']
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
M^2PX \arrow[r, "\lambda^2_X"] \arrow[d, "iP^2_X"'] \arrow[dr, phantom, "\Downarrow(3)"] & PM^2X \\
S^2PX \arrow[r, "\rho^2_X"] \arrow[d, "m_{PX}"'] \arrow[dr, phantom, "\Downarrow(4)"] & PS^2X \arrow[u, "\kappa_{m_X}"'] \\
SPX \arrow[r, "\rho_X"'] & PSX \arrow[u, "\kappa_{i_X}"']
\end{tikzcd}$$

We begin with an analysis of the left-hand-side. For readability, we write $(f_1^n) := f_1 \otimes \ldots \otimes f_n$.

The top passage is given by

$$\begin{aligned}
(f_1^{k_1}) \otimes \ldots \otimes (f_1^{k_n}) &\overset{M\lambda_X}{\mapsto} (f_1^{k_1})(-) \times \ldots \times (f_1^{k_n})(-) \\
&\overset{\lambda_{MX}}{\mapsto} \big(f_1(-) \times \ldots \times f_{k_1}(-)\big) \times \ldots \times \big(f_1(-) \times \ldots \times f_{k_n}(-)\big)
\end{aligned} \tag{9.5}$$

the bottom passage sends

$$
\begin{aligned}
(f_1^{k_1}) \otimes \ldots \otimes (f_1^{k_n}) &\overset{M_{PX}}{\mapsto} f_{11} \otimes \ldots \otimes f_{1k_1} \otimes \ldots \otimes f_{n1} \otimes \ldots \otimes f_{nk_n} \\
&\overset{i_{PX}}{\mapsto} f_{11} \otimes \ldots \otimes f_{1k_1} \otimes \ldots \otimes f_{n1} \otimes \ldots \otimes f_{nk_n} \\
&\overset{\rho_X}{\mapsto} \int^{x_{11},\ldots,x_{nn_k} \in X} f_{11}(x_{11}) \times \ldots \times f_{nn_k}(x_{nn_k}) \times SX(-, x_{11} \otimes \ldots \otimes x_{nk_n}) \\
&\overset{\kappa_{i_X}}{\mapsto} \int^{x_{11},\ldots,x_{nn_k} \in X} f_{11}(x_{11}) \times \ldots \times f_{nn_k}(x_{nn_k}) \times SX(i_X(-), x_{11} \otimes \ldots \otimes x_{nk_n}) \\
&\overset{\kappa_{i_X}}{\mapsto} \int^{x_{11},\ldots,x_{nn_k} \in X} f_{11}(x_{11}) \times \ldots \times f_{nn_k}(x_{nn_k}) \times SX(m_X(-), x_{11} \otimes \ldots \otimes x_{nk_n})
\end{aligned} \tag{9.6}
$$

It is not difficult to see that (1) just removes the inner bracketing from the expression in 9.5 and (2) embeds this expression into 9.6. A similar story repeats itself on the right-hand-side: (3) is an embedding and (4) removes brackets. Hence the two passages are indeed equal.

9.1.3 Applying Theorem 1 to i

In the previous two sections we showed that $i : M \hookrightarrow S$ is a monad morphism in DL^P. Thus by Chapter 7 it can be extended to a monad morphism

$$i_* : \tilde{M} \to \tilde{S}.$$

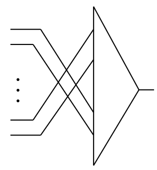
Applying Theorem 8.1 of Chapter 8 we see that there is a 2-functor

$$I : \mathsf{kl}(\tilde{M}) \to \mathsf{kl}(\tilde{S}).$$

Since this is a 2-functor it sends monads to monads and distributive laws to distributive laws (cf. Chapter 6). Let $\varphi : 1 \nrightarrow 1$ be a monad in $\mathsf{kl}(\tilde{M})$. Then

$$
\begin{aligned}
I(\varphi) = {}& 1 \xrightarrow{\varphi} M1 \xrightarrow{(i_1)_*} S1 \\
= {}& \int^{n \in \mathbb{N}} \varphi(n, 1) \times S1(-, n)
\end{aligned} \tag{9.7}
$$

Operations of arity n in $I(\varphi)$ consist of operations of arity n in φ precomposed with symmetries. In other words, I freely adds symmetry to all the operations of φ. We visualise these operations as follows:



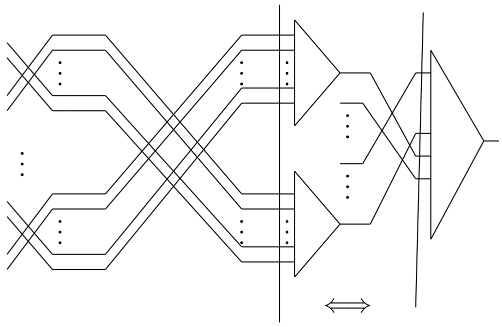
The big question is how does can we compose these new operations? Composition given by

$$I(\varphi) \cdot I(\varphi) \overset{\cong}{\Rightarrow} I(\varphi \cdot \varphi) \to I(\varphi)$$

We compute

$$
\begin{aligned}
I(\varphi)\cdot I(\varphi)(n) &= \big(m_1\cdot \tilde{S}(i_1)_*\cdot \tilde{S}\varphi\cdot (i_1)_*\cdot\varphi\big)(r)\\
&\simeq \int^{n'\in M1}\int^{n\in S1}\int^{k_1\otimes\ldots\otimes k_n\in SM1}\int^{\sum k'_i=n}\varphi(n')\times S1(n,n')\times \tilde{S}\varphi(k_1\otimes\ldots\otimes k_n,n)\times\prod_1^r S1(k'_i,k_i)
\end{aligned}
$$

which should be interpreted as



Distiribute

To compose the above operation with the rules we have, we need to distribute the operations in $\tilde{S}(\varphi)$ over the symmetries. To understand how this is done, we need to carefully examine the isomorphism $I(\varphi)\cdot I(\varphi)\overset{\simeq}{\to} I(\varphi\cdot\varphi)$. This isomorphism is given by the 2-cell (cf. Theorem 8.1)

$$
\begin{tikzcd}
1 \arrow[r, "\varphi", "\shortmid" marking] & M1 \arrow[r, "(i_1)_*", "\shortmid" marking] \arrow[dr, "\tilde{M}\varphi"', "\shortmid" marking] & S1 \arrow[r, "\tilde{S}\varphi", "\shortmid" marking] \arrow[d, phantom, "\Downarrow a"] & SM1 \arrow[r, "\tilde{S}(i_1)_*", "\shortmid" marking] & S^2 1 \arrow[r, "m_1", "\shortmid" marking] & S1 \\
& & M^2 1 \arrow[r, "\tilde{M}(i_1)_*"', "\shortmid" marking] \arrow[ur, "(i_{M1})_*"', "\shortmid" marking] \arrow[dr, "m_1"', "\shortmid" marking] & MS1 \arrow[ur, "(i_{S1})_*"', "\shortmid" marking] & & \\
& & & M1 \arrow[uurr, "(i_1)_*"', "\shortmid" marking] & &
\end{tikzcd}
$$

Distributivity is achieved by the 2-cell a which is explicitly given by the following isomorphism

$$
\int^{n\in S1}\tilde{S}\varphi(k'_1\otimes\ldots\otimes k'_n,n)\times S1(n,n')\to\tilde{S}\varphi(k'_1\otimes\ldots\otimes k'_n,n');
$$

Fix $n\in S1$. A typical element of the left-hand-side above is the triple

$$
\big(((x_1,\ldots,x_n),\tau),\sigma\big),
$$

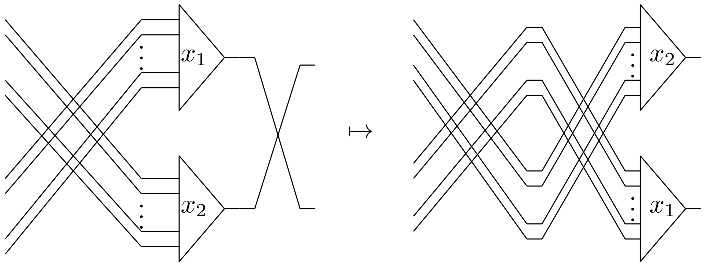
where $\tau : k'_1\otimes\ldots\otimes k'_n\to k_1\otimes\ldots\otimes k_n$ is a symmetry isomorphism, as is $\sigma : n\to n'$. To get to the right-hand-side, we send this triple to

$$
\tilde{S}\varphi(k'_1\otimes\ldots\otimes k'_n,\sigma)\big((x_1,\ldots,x_n),\tau\big)=\big((x_{\sigma(1)},\ldots,x_{\sigma(n)}),\tau'\big)
$$

with

$$
\tau' := \ k'_1\otimes\ldots\otimes k'_n\xrightarrow{\ \tau\ } k_1\otimes\ldots\otimes k_n\longrightarrow k_{\sigma(1)}\otimes\ldots\otimes k_{\sigma(n)}\ .
$$

This process is visualised in the following figure:



The main point is that the operations of the new theory $I(\varphi)$ are composed exactly like the operation of the original theory φ. The only difference is that in $I(\varphi)$ we first need to distributive the operations over symmetries in order to be able to exploit the composition in φ. Once we distributed, we know exactly how to compose both the operations of φ and the symmetries.

9.1.4 Applying Theorem 2 to i

In this section we try to explain why Theorem 2 applies to $i : M \hookrightarrow S$. Doing so involves performing numerous technical calculations with coends. Since coend manipulation methods were explored in detail in the previous section, the proofs in this section are of a more conceptual nature. They explain why our arguments are correct, but do not show this correctness in full detail. We hope the reader will welcome this refreshing change.

The first assumption we make is that all our categories are discrete. This hypothesis does not specialise our arguments because they rely on taking coends of MX and SX and so if X were not discrete, the non-trivial morphisms of X would be quotiented out by the coend.

To show applicability of Theorem 2 we need to first show that the family $(i_X)^*$ defines a 2-monad morphism. This will involve building several 2-cells and verifying coherence for them. Doing this requires us to show how these 2-cells act on elements of coends. To simplify our discussion, we shall use graphical notation to both define the elements of the coends and to define how our 2-cells will act on these elements. We begin by illustrating this with a detailed example.

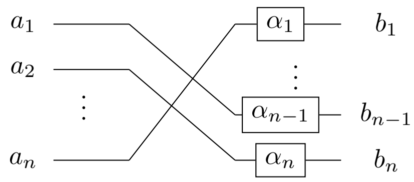
Our first task is to show that the family $(i_X)^*$ is a 2-natural transformation. To this end, we need, for every profunctor $\varphi : x \nrightarrow y$, to construct the 2-cell

$$\begin{tikzcd} Sx \arrow[r, "\tilde{S}\varphi", "\shortmid"{marking}] \arrow[d, "(i_x)^*"', "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow\iota_\varphi"] & Sy \arrow[d, "(i_y)^*", "\shortmid"{marking}] \\ Mx \arrow[r, "\tilde{M}\varphi"', "\shortmid"{marking}] & My \end{tikzcd} \tag{9.8}$$

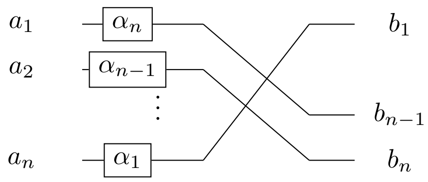
The first step in this construction is to evaluate the top-right and the bottom-left passages in the diagram. For example, the top right passage evaluated to

$$((i_y)^* \cdot \tilde{S}\varphi)(a_1 \otimes \ldots \otimes a_n, b_1 \otimes \ldots \otimes b_n) = \int^{c_1 \otimes \ldots \otimes c_n \in Sy} Sy(a_1 \otimes \ldots \otimes a_n, c_1 \otimes \ldots \otimes c_n) \times \tilde{S}\varphi(c_1 \otimes \ldots \otimes c_n, b_1 \otimes \ldots \otimes b_n) \tag{9.9}$$

instead of dealing with this expression directly, we pick a typical element in the right-hand-side of 9.9:



where for every i, $\alpha_i \in \varphi(a_i, b_i)$. We can similarly represent the left-hand-side:

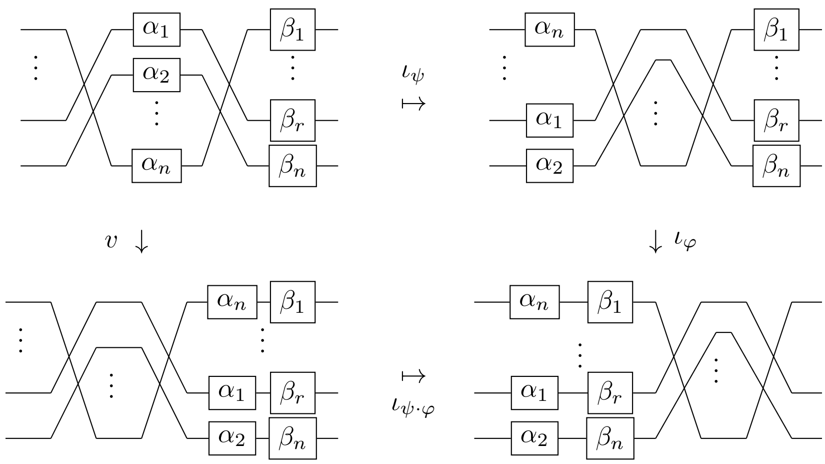


The 2-cell of diagram 9.8 is given by sliding the boxes from right to left in the above diagrams.

The graphical view makes it easy to verify that coherence holds. We check that for profunctors $\varphi : x \nrightarrow y$ and $\psi : y \nrightarrow z$

$$
\begin{tikzcd}
 & & Sy \arrow[drr, "\tilde{S}\psi"] \arrow[dd, "(i_y)^*"] & & \\
Sx \arrow[urr, "\tilde{S}\varphi"] \arrow[ddd, "(i_x)^*"'] \arrow[r, phantom, "\Downarrow\iota_\varphi"] & & & \arrow[r, phantom, "\Downarrow\iota_\psi"] & Sz \arrow[ddd, "(i_z)^*"] \\
 & & My \arrow[ddrr, "\tilde{M}\psi"] \arrow[d, phantom, "="] & & \\
 & & {} & & \\
Mx \arrow[uurr, "\tilde{M}\varphi"] \arrow[rrrr, "\tilde{M}(\psi\cdot\varphi)"'] & & & & Mz
\end{tikzcd}
=
\begin{tikzcd}
 & Sy \arrow[dr, "\tilde{S}\psi"] \arrow[d, phantom, "\Downarrow v"] & \\
Sx \arrow[ur, "\tilde{S}\varphi"] \arrow[rr, "\tilde{S}(\psi\cdot\varphi)"] \arrow[d, "(i_x)^*"'] & \arrow[d, phantom, "\Downarrow\iota_{\psi\cdot\varphi}"] & Sz \arrow[d, "\alpha_Z"] \\
Mx \arrow[rr, "\tilde{M}(\psi\cdot\varphi)"'] & {} & Mz
\end{tikzcd}
$$

which follows from the following equality

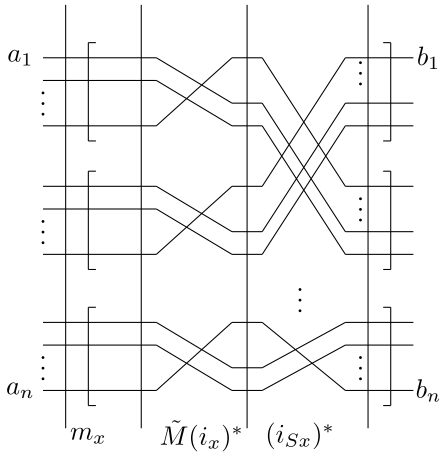


The other two conditions for 2-natural transformations follows by similar arguments.

Next, we show that i^* is a monad morphism. The 2-cell

$$
\begin{tikzcd}
S^2x \arrow[r, "(i_{Sx})^*"] \arrow[d, "m_x"'] & MSx \arrow[r, "\tilde{M}(i_x)^*"] & M^2x \arrow[d, "m_x"] \\
Sx \arrow[rr, "(i_x)^*"'] & & Mx
\end{tikzcd}
$$

is given by sending



to its composite in $Sx(a_1 \otimes \ldots \otimes a_n, b_1 \otimes \ldots \otimes b_n)$. (Note that we used short vertical arrows in the diagram to denote the partitions of the lists $a_1 \otimes \ldots \otimes a_n$ and $b_1 \otimes \ldots \otimes b_n$ i.e. lists of short arrows denote elements of S^2x and M^2x).

One can show that the coherence diagrams (cf. Equations **??**) hold. For example, a typical element of the the top-right passage consists of several types of permutations (permutations between objects of x, objects of Mx etc). We know that each 2-cell that makes up the both the right- and the left-hand side either composes permutations or regroups the permuted terms. Since permutations compose uniquely, both passages send the input data to the same permutation and are hence equal.

The second structural 2-cell

$$\begin{tikzcd} x \arrow[r, "u_x", "\shortmid"{marking}] \arrow[dr, "u_x"', "\shortmid"{marking}] & Sx \arrow[d, "(i_x)^*", "\shortmid"{marking}] \\ & Mx \end{tikzcd}$$

is strict since both passages in the above diagram evaluate to

$$u_x(a, b) = X(a, b) = (i_x)^* \cdot u_x(a, b).$$

We conclude

Proposition 9.3. *The family $(i_x)^*$ forms a monad morphism $i^* : \tilde{S} \to \tilde{M}$.*

Hence by Theorem **??**, i^* gives rise to a 2-functor

$$J : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{M}).$$

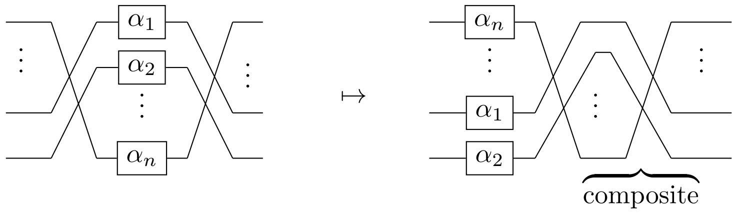
Let $\varphi : 1 \nrightarrow 1$ be a monad in $\mathsf{kl}(\tilde{S})$ we compute

$$J(\varphi)(n, 1) = \int^{m \in S1} \varphi(m, 1) \times S1(n, m) \simeq \varphi(i_1(n), 1).$$

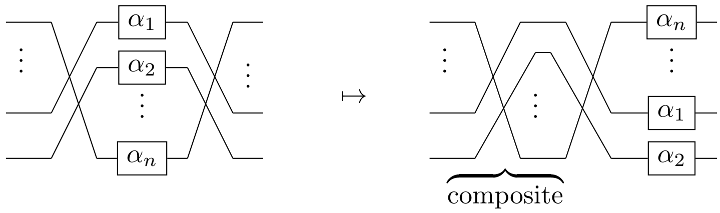
We see that $J(\varphi)$ forgets the action of φ on the symmetries.

Our next goal is to make the forgetful-free relationship between I and J precise. To do this, we will apply Theorem 8.3 to i. We check that the pair i_*, i^* satisfied conditions 8.6 and 8.7. The other two conditions are can be proved similarly, albeit using slightly more elaborate calculations.

We represent condition 8.6 graphically. The left-hand-side sends



with the rightmost symmetry a compose in $S1(x_1 \otimes \ldots \otimes x_n, y_1 \otimes \ldots \otimes y_n)$. The right-hand-side sends



And these two morphisms are equal in $S1(x_1 \otimes \ldots \otimes x_n, y_1 \otimes \ldots \otimes y_n)$.

Condition 8.7 is easier. Both diagrams just embed $y(y_0, y_1)$ into $\{*\} \times y(y_0, y_1)$.

Hence the Theorem allows to conclude the theres exists a lax 2-adjunction $J \dashv I$ and hence in particular an adjunction

$$J_{1,1} \dashv I_{1,1} : \mathsf{kl}(\tilde{M})(1,1) \to \mathsf{kl}(\tilde{S})(1,1)$$

that allows us to construct symmetric operads from plain ones and to view symmetric operads as plain ones by forgetting the action of symmetries.

9.2 Example 2: $j : S \hookrightarrow F$

In this section we show that the inclusion of symmetric monoidal categories into finite-product categories induces a lax 2-adjunction

$$\begin{tikzcd} \mathsf{kl}(\tilde{S})(1,1) \arrow[r, shift left] \arrow[r, phantom, "\top"] & \mathsf{kl}(\tilde{F})(1,1) \arrow[l, shift left] \end{tikzcd}$$

We do this in several steps. First we explain why j is a 1-cell in DL^P. Then we show that j is a monad morphism in DL^P. These two results allows us to apply Theorem 8.1 to j to produce a 2-functor

$$K : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{F}).$$

Next we show that $j^* : \tilde{F} \to \tilde{S}$ is also a monad morphism. Applying Theorem 8.1 for the second time, we obtain a 2-functor in the other direction

$$K' : \mathsf{kl}(\tilde{F}) \to \mathsf{kl}(\tilde{S}).$$

We then verify that j^* and j^* satisfy the conditions of Theorem **??**, giving us the desired result.

Note that most of the work was already done in the previous section for $i : M \hookrightarrow S$. The arguments given there are isomorphic to the arguments we need to give for j. Here we construct the main ingredients needed for the formal proofs, and then refer the reader to Sections 9.1.1 – 9.1.4 for details on how to implement them.

To show that j is a 1-cell in DL^P we need to construct the 2-cell

$$\begin{tikzcd} SP \arrow[r, "\rho"] \arrow[d, "jP"'] \arrow[dr, phantom, "\Downarrow\hat{j}"] & PS \arrow[d, "Pj"] \\ FP \arrow[r, "\xi"'] & PF \end{tikzcd}$$

We evaluate both passages in the diagram. The top-right passage is given by

$$\begin{aligned} f_1 \otimes \ldots \otimes f_n &\mapsto \int^{x_1,\ldots,x_n \in X} f_1(x_1) \times \ldots \times f_n(x_n) \times SX(-, x_1 \otimes \ldots \otimes x_n) \\ &\mapsto \int^{y_1,\ldots,y_p \in SX} \int^{x_1,\ldots,x_n \in X} f_1(x_1) \times \ldots \times f_n(x_n) \times SX(y_1 \otimes \ldots \otimes y_p, x_1 \otimes \ldots \otimes x_n) \\ &\qquad FX(-, y_1 \times \ldots \times y_p) \\ &\simeq \int^{x_1,\ldots,x_n \in X} f_1(x_1) \times \ldots \times f_n(x_n) \times FX(-, x_1 \times \ldots \times x_n) \end{aligned} \tag{9.10}$$

while the bottom-left passage computes to

$$\begin{aligned} f_1 \otimes \ldots \otimes f_n &\mapsto f_1(-) \times \ldots \times f_n(-) \\ &\mapsto \int^{x_1,\ldots,x_n \in X} f_1(x_1) \times \ldots \times f_n(x_n) \times FX(-, x_1 \times \ldots \times x_n) \end{aligned} \tag{9.11}$$

Hence $\hat{j}$, much like i^* of Section 9.1.1, sends elements of 9.10 to themselves in 9.11. Because of this, showing coherence for $\hat{j}$ is almost identical to showing coherence for i^*. The only difference is that the actions of symmetries is replaced by actions of finite-product structure, but this does not alter the nature of the argument.

An argument identical to the one given in Section 9.1.2 shows that $(j, \hat{j})$ is in fact a monad morphism in DL^P. Hence $j_* : \tilde{S} \to \tilde{F}$ is also a monad morphism and Theorem **??** can be invoked to produce a 2-functor $K : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{F})$. In particular, K sends monads in $t : 1 \nrightarrow 1$ in $\mathsf{kl}(\tilde{S})$ to monads $K(t) : 1 \nrightarrow 1$ in $\mathsf{kl}(\tilde{F})$. We compute

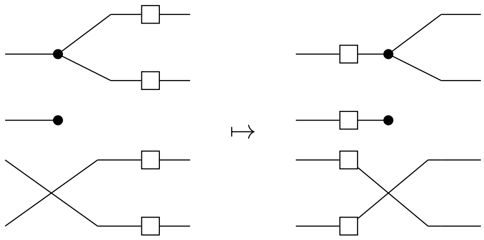
$$K(t)(n,1) = \int^{k \in S1} t(k,1) \times F1(n,k)$$

While the above formula is structurally very similar to Equation 9.7, there is one key difference. The above coend is taken over all $k \in S1$ and hence the symmetries in $t(k,1)$ and the symmetries in $F1(n,k)$ are identified. This feature is essential for the formula to make sense. Hence $K(t)$ consists off all the operations of t to which we formally adjoin finite-product structure, but in a way that identifies the symmetries of the finite product structure with the existing symmetries of t.

Furthermore it turns out that $j^* : \tilde{F} \to \tilde{S}$ is also a monad morphism. The definition of the 2-cell

$$\begin{tikzcd} FX \arrow[r, "\tilde{F}\varphi", "\shortmid" marking] \arrow[d, "(j_x)^*"', "\shortmid" marking] \arrow[dr, phantom, "\Downarrow"] & Fy \arrow[d, "(j_y)^*", "\shortmid" marking] \\ Sx \arrow[r, "\tilde{S}\varphi"', "\shortmid" marking] & Sy \end{tikzcd}$$

can be derived from



Arguments similar to the ones gives in Section 9.1.4 can be used to show that $j^* : \tilde{F} \to \tilde{S}$ is indeed a monad morphism. By Theorem **??** this gives rise to a 2-functor

$$K' : \mathsf{kl}(\tilde{F}) \to \mathsf{kl}(\tilde{S}).$$

Given a monad $t : 1 \nrightarrow 1$ in $\mathsf{kl}(\tilde{F})$, we compute

$$K'(t)(n,1) = \int^{k \in F1} t(k,1) \times F1(i(n),k) \simeq t(i(n),1)$$

One can give a nice interpretation to the formulas for $K'(t)$: it is the theory that contains all the operations of t, but forgets some actions of finite-product structure on these operations. Crucially, the actions of the symmetries are not forgotten, while the actions of copying and deletion operations are. Thus $K(t')$ is indeed a monad in $\mathsf{kl}(\tilde{S})$.

A graphical argument, akin to the one given in Section 9.1.4, shows that j^* and j_* satisfy the conditions the Theorem **??** and this allows us to conclude that there exists a lax 2-adjunction

$$K' \dashv K : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{F}),$$

as promised.

Chapter 10

Conclusions and future work

10.1 Summary

Let us review the main results of this work. We began with an introduction to Categorical Universal Algebra. This field uses various categories and category-like structures to capture the syntax of algebraic theories. We saw two main variants. PROs, PROPs and Lawvere theories are categories whose morphisms fall into two groups. The structure morphisms of the category - such as symmetries or finite product structure - and the morphisms that encode the specifications of the theory. The second variant was operad-based theories. These model specifications in which the operations have many inputs, but only a single output. We encountered many types of operads. Specifically monoidal, symmetric monoidal and Cartesian operads. We concluded that there is a need for a setting in which will allow us to formulate and analyse theories in general, without worrying about the complications associated with types.

In Chapters 4 and 5 we developed the 2-categorical foundations for such a framework. All the main definitions were accompanied by numerous examples. Of particular importance were the Cat-monads M, S, F and P. The first three turn a category X into the free monoidal category MX, the free symmetric monoidal category SX and the free finite product category FX. P is the presheaf monad and is the categorification of the powerset monad on Set. One of the the most important facts about it is that its Kleisli bicategory is equivalent the the bicategory Prof of profunctors.

2-distributive laws consist of a 2-natural transformation $\lambda : TP \to PT$. Through Beck's Theorem, they provide us with a way of extending T to the Kleisli bicategory of P and lifting P to the category of T-algebras. We showed that M, S and F all distribute over P and hence extend to Prof. Finally, we described the formulas for these extensions.

Chapter 6 showed how the bicategory of profunctors, and its variations, can be used to formulate constructions from Categorical Universal Algebra. We began with a simple, well-know, example: categories are monads in Span. From there we studied monads in Prof. It turned out that a monad on X in Prof adds categorical structure to X in a way that makes that structure compatible with the existing structure of X. We observed that this view point is useful for defining PROPs and Lawvere theories because it allows one to separate the structural morphisms of the type from the operations of the theory that one wishes to encode. The monoidal structure of PROPs led to us to work in Prof(Mon) instead of Prof.

Because operads model theories with multiple inputs, but only a single output, they no longer can be modelled as morphisms in a monoidal category. Instead they are morphisms in muticategories. Hom-sets of multicategories are functors

$$\hom : TX \times X \to \mathsf{Set}$$

with T some monad on Cat. We observed that this is the same as a 1-cell in the Kleisli bicategory $\mathsf{kl}(\tilde{T})$ with $\tilde{T}$ an extension of T to Prof. Our running examples - M, S and F - can all be extended to Prof. This allowed us to study monads in $\mathsf{kl}(\tilde{M})$, $\mathsf{kl}(\tilde{S})$ and $\mathsf{kl}(\tilde{F})$. We showed that monads on 1 in these bicategories correspond to plain, symmetric and Cartesian operads. The Author learned about the last set of results from Martin Hyland, but has not seen them in literature. He hopes that by clearly documenting them in this work, he will aid in their public recognition.

We also discussed distributive laws between monads inside Kleisli bicategories. By viewing theories as monads, we get a compositional tool for free. We showed interesting examples of distributive laws in Prof. Through these laws one obtains a particularly simple description of the PROPs for bialgebras and Frobenius algebras. We also gave an example of a distributive law between the theory for semi-groups and the theory for pointed sets in $\mathsf{kl}(\tilde{S})$.

Chapters 7, 8 and 9 contain the original results of this work. The aim of these chapters is to show that relationships between types leads to relationships between theories of these types. Restating this formally, we wish to begin with a monad morphism

$$\alpha : S \to T$$

from a types S to a type T and from it to construct a 2-functor

$$A : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{T}).$$

This is done in two steps. First we wish to extend the monad morphism $\alpha : S \to T$ to a monad morphism $\tilde{\alpha} : \tilde{S} \to \tilde{T}$ and then show that post-composing with this extension is 2-functorial.

Chapter 7 began as a single theorem that showed that certain α's can indeed be extended. But the author quickly realised that it is worthwhile to explore the theory of extensions in general, rather than focusing only on a single aspect of it. This led to the creation of two monoidal bicategories, DL^P and Ext_P. The first one contains 2-functors, 2-natural transformations and modifications that distribute over P. The second contains the all 2-functors, 2-natural transformations and modifications that can be extended to $\mathsf{kl}(P)$. Chapter 7 carefully builds an equivalence between these two bicategories. The resulting formal theory is independent from the application we use it for. Indeed, Chapter 7 is self-contained a result in 2-monad theory. It complements the work of Tanaka who showed in [Tanaka, 2005] that a similar equivalence holds between a bicategory dual to DL^P and the 2-category of lifts of 2-functors, 2-natural transformations and modifications to the bicategory of algebras of P.

Chapter 8 brings us closer to Categorical Universal Algebra, although the two Theorems that we prove there are again self-contained. The first theorem shows that any monad morphism $\alpha : S \to T$ gives rise to a 2-functor $A : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(T)$. We can combine this result with the results of Chapter 7 to see that any monad morphism $\alpha : S \to T$ that is also a 1-cell in DL^P gives rise to such a 2-functor. This 2-functor sends theories of type S to theories of type T. This holds true for any P, not just the presheaf monad. When P is the presheaf monad something more exciting happens. Any functor f gives rise to two profunctors: f^* and f_*.

Hence any 1-cell $\alpha : S \to T$ in DL^P gives rise to two collections

$$\{(a_X)_* \mid X \in \mathsf{Cat}\} \text{ and } \{(a_X)^* \mid X \in \mathsf{Cat}\}$$

We know that the one on the left is a 2-natural transformation. If we assume that the one on the right is a monad morphism, then by the first Theorem of Chapter 8, we obtain a functor $J : \mathsf{kl}(\tilde{T}) \to \mathsf{kl}(\tilde{S})$.

Because of $f_* \dashv f^*$ we expect the two functors A and J to be related. Theorem 2 of Chapter 8 makes this relationship explicit. It states that when the units and co-units of the family of adjunctions $(\alpha_y)_* \dashv (\alpha_y)^*$ satisfy some coherence conditions, J and A form a lax 2-adjunction.

In Chapter 9 we apply the results of Chapters 7 and 8 to the inclusions $M \hookrightarrow S$ and $S \hookrightarrow F$. This requires us to check numerous technical conditions, but once we overcome this hurdle, we show that the inclusions produce adjunctions

$$\mathsf{kl}(\tilde{M})(X,X) \underset{\longleftarrow}{\overset{\longrightarrow}{\top}} \mathsf{kl}(\tilde{S})(X,X) \qquad \text{and} \qquad \mathsf{kl}(\tilde{S})(X,X) \underset{\longleftarrow}{\overset{\longrightarrow}{\top}} \mathsf{kl}(\tilde{F})(X,X).$$

The first adjunction specifies how to create free symmetric operads from a given plain operad and how to forget the symmetric structure of a symmetric operad. The second one does the same for symmetric operads and Lawvere theories. Furthermore, the 2-functors that make up the lax 2-adjunction give a recipe for creating a free distributive law between Lawvere theories from a given distributive laws between symmetric operads and this last corollary was one of the initial motivations for pursuing the research program in this thesis.

10.2 Future work

This thesis developed a method for extending relationships between types to relationships between theories of these types. While this method works well, we believe that it can be greatly enhanced with further investigation. The main deficiency of the current approach can be seen in Chapter 9. To apply our results to the monad morphism $i : M \hookrightarrow S$, we had to verify numerous non-trivial technical conditions. We believe that things should be simpler. To this end, we propose the following research programme:

- **Identify a class of monad morphisms $i : S \to T$ that can be extended to $\tilde{i} : \tilde{S} \to \tilde{T}$.** We suspect that inclusions, such as the examples we explored in Chapter 9, can always be extended. We suspect that a careful analysis of the arguments used to show that $i : M \hookrightarrow S$ in a 1-cell in DL^P will help identify the essential ingredients needed for a monad morphisms to be extendible.
- **Identify which $i : S \to T$ are such that $i^* : T \to S$ is a natural transformation**. Based on our limited class of examples, inclusions might be a good candidate to investigate first. We also suspect that the monads S and T need to possess some nice properties for i^* to be a 2-natural transformation. We think that this question is the most important theoretical questions that is left unexplored in this thesis.

The thesis also opens the door to many new questions in Categorical Universal Algebra.

- **How do PROs and PROPs fit in?** PROs and PROPs provide a simple semantics for capturing the syntax of algebraic theories. Yet they do not fit into our Kleisli bicategorical framework. The problem is that a 1-cell in $\mathsf{kl}(T)$ has the form $X \nrightarrow TY$, while to model several outputs, we would need to work with a 1-cell $TX \nrightarrow TY$. There are several ways to accomplish this. One is to define a monad $R : \mathsf{kl}(T) \to \mathsf{kl}(T)$ and then study its co-Kleisli category. There a typical one cell will have the form $RX \nrightarrow TY$. To model PROPs we then need to extend the 2-monad $\tilde{S}$: Prof $\to$ Prof to $\tilde{\tilde{S}} : \mathsf{kl}(\tilde{S}) \to \mathsf{kl}(\tilde{S})$. For this we need yet another 2-distributive law. Richard Garner's Thesis [Garner, 2006] defines polycategories as monads in a two-sided Kleisli bicategory. This bicategory is obtained from a 2-distributive law of a 2-monad T on Prof over the extension $\tilde{S}$: Prof $\to$ Prof. The idea that multi-categories are monads in $\mathsf{kl}(\tilde{S})$ serves as a foundation for Garner's work. Because of this, the tools and ideas developed in this thesis can be imported to his two-sided setting.

- **Comparison with the** Prof(Mon) **setting.** [Lack, 2004] develops an elegant setting for composing PROs and PROPs. In [Akhvlediani, 2010] the author shows that Lawvere theories and so-called PROBs can also be composed in Prof(Mon). We believe that it is important to understand how the Kleisli bicategory framework compares to the Prof(Mon) setting. For example, one advantage of the latter is that it captures theories with multiple outputs and provides explicit formulas for describing important distributive laws between them. Can such distributive laws be described in the Kleisli setting? Or, more realistically, can they be described in the two-sided Kleisli setting ala Garner? Certainly the Kleisli setting has more expressing power, since it is not restricted to theories with linear arities. But when it comes to linear arities, are the expressive powers of the two setting compatible?
- **Examples.** We have presented an abstract high-level setting. It is a prototype of the theory we wish to eventually develop. But to develop it, we need to test it out with examples. There are two things we wish to have more examples of. First, we would like to see more theories expressed as monads in $\mathsf{kl}(\tilde{T})$. In particular, it would be interesting to see whether there are familiar categorical constructions that happen to be monads in $\mathsf{kl}(\tilde{T})$. This is certainly the case with PROPs and PROs where familiar categories happen to capture the syntax of well-known algebraic structures. Second, we would like to have more examples of types. At this point we only have three running examples: M, S and F. This make it difficult to see the full expressive power of the Kleisli bicategory setting. We believe that finite products can be replaced with more elaborate limits to produce types that were previously not studied. Ideally, both new theories and new types will come from an exciting demand in computer science to model parts of programming languages using algebraic specifications. For example, monadic effect in functional programming languages, such as Haskel, can be modelled using a variant of Lawvere theories (cf. [Hyland and Power, 2006]). Representing such concrete Lawvere theories as monads in a Kleisli bicategory would not only give us insight into our theory, but would also give rise to a coveted (cf. [Hyland et al., 2002]) method for combining such computational effects.
- **Incorporating other approaches to Universal Algebra.** Categorical Universal Algebra contains more tools than we could explore in this thesis. Sketches are effectively used to describe algebraic structure in Computer Science [Barr and Wells, 1985]. Briefly, a *sketch* is a category with a chosen set of limit cones and colimit cocones. Models of a sketch are functors out of the sketch that map cones and cocones to limits and colimits in the codomain. Since sketches are categories with extra structure, they too should be expressible in a variant of the bicategory of profunctors. There exist many different types of sketches. Finite product sketches are discussed in [Barr and Wells, 1985] and [Barr and Wells, 1993]; symmetric monoidal sketches and monoidal sketches are covered in [Hyland and Power, 2000]. The Kleisli bicategory framework should allow all these flavours to be expressed in a single setting. Furthermore, expressing sketches, PROs and operads using the same language will make it easier to analyse the expressive power of each and to compare the strengths and weaknesses of these tools.
- **Applications to Categorical Quantum Mechanics.** [Abramsky and Coecke, 2004] developed category

theoretic semantics for some quantum informatics protocols, most notably Quantum Teleportation. The research program initiated by that paper led to many fascinating results that advanced our understanding of both Category Theory and Quantum Foundations. Many of these results rely on a representation of quantum observables as so-called †-Frobenius Algebras [Coecke and Pavlovic, 2007]. Furthermore, interacting observables can be expressed as two †-Frobenius algebras that satisfy the axioms of [Coecke and Duncan, 2008]. Since the PROP for Frobenius algebras can be expressed as a monad in Prof it can be combined with other PROPs in that setting. This opens interesting possibilities for semantics of quantum information. First, distributive laws in Prof can suggest new ways of combining Frobenius algebras. It would be interesting to see if the resulting structures are meaningful in physics. Furthermore, PROPs for Frobenius algebras can be combined with other PROPs. Through such combinations, one can create semantics for rather complicated systems. Finally, because both Lawvere theories and PROPs can be expressed as monads in Prof, they can, modulo some technical details, be combined with each other. This could lead to category theoretic semantics for a systems in which classical and quantum protocols interact.

Bibliography

[Abramsky and Coecke, 2004] Abramsky, S. and Coecke, B. (2004). A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium of Logic in Computer Science*, pages 415–425. IEEE Computer Science Press. Extended version: arXiv:0808.1023.

[Adamek et al., 1990] Adamek, J., Herrlich, H., and Strecker, G. (1990). *Abstract and concrete categories: The joy of cats*. John Wiley and Sons.

[Akhvlediani, 2009] Akhvlediani, A. (2009). Transfer report.

[Akhvlediani, 2010] Akhvlediani, A. (2010). Composing lawvere theories. *CT2010 Presentation*.

[Barr and Wells, 1985] Barr, M. and Wells, C. (1985). *Toposes, Triples and Theories*. Springer-Verlag.

[Barr and Wells, 1993] Barr, M. and Wells, C. (1993). *Category Theory for Computing Science*.

[Beck, 1969] Beck, J. (1969). Distributive laws. *Lecture Notes in Mathematics*, 80(119-140).

[Bénabou, 1967] Bénabou, J. (1967). Introduction to bicategories. In *Reports of the Midwest Category Seminar*, volume 47 of *Lecture Notes in Mathematics*, pages 1–77. Springer Berlin / Heidelberg. 10.1007/BFb0074299.

[Blackwell et al., 1989] Blackwell, R., Kelly, G. M., and Power, A. (1989). Two-dimensional monad theory. *Journal of Pure and Applied Algebra*.

[Cheng, 2011] Cheng, E. (2011). Distributive laws for lawvere theories.

[Cheng et al., 2004] Cheng, E., Hyland, M., and Power, J. (2004). Pseudo-distributive laws. *Electornic Notes in Theoretical Computer Science*, 83.

[Coecke and Duncan, 2008] Coecke, B. and Duncan, R. (2008). Interacting quantum observables. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science. Extended version: arXiv:quant-ph/09064725.

[Coecke and Pavlovic, 2007] Coecke, B. and Pavlovic, D. (2007). Quantum measurements without sums. In Chen, G., Kauffman, L., and Lamonaco, S., editors, *Mathematics of Quantum Computing and Technology*, pages 567–604. Taylor and Francis. arXiv:quant-ph/0608035.

[Curien, 2008] Curien, P.-L. (2008). Operads, clones, and distributive laws.

[Ehresmann, 1965] Ehresmann, C. (1965). *Categories et Structures*. Dunod.

[Gambino, 2009] Gambino, N. (2009). On the coherence condition for pseudo-distributive laws.

[Garner, 2006] Garner, R. (2006). *Polycategories*. PhD thesis, Cambridge University.

[Gordon et al., 1995] Gordon, R., Power, A., and Street, R. (1995). Coherence of tricategories. *Mem. Amer. Math. Soc.*, 558.

[Gray, 1974] Gray, J. (1974). *Formal Category Theory: Adjointness for 2-Categories*. Springer Lecture Notes in Mathematics.

[Gurski, 2007] Gurski, N. (2007). *An algebraic theory of tricategories*. PhD thesis, University of Chicago.

[Houston, 2008] Houston, R. (2008). *Linear Logic without Units*. PhD thesis, University of Manchester.

[Hyland, 2010] Hyland, M. (2010). Talk at the categories, logic and physics. *Oxford*.

[Hyland et al., 2002] Hyland, M., Plotkin, G., and Power, J. (2002). Combining computational effects: commutativity and sum. *Proceedings of IFIP TCS'2002*.

[Hyland and Power, 2000] Hyland, M. and Power, J. (2000). Symmetric monoidal sketches. *Proceedings of the 2nd ACM SIGPLAN international conference on Principles and practice of declarative programming*.

[Hyland and Power, 2006] Hyland, M. and Power, J. (2006). Discrete lawvere theories and computational effects. *Theoretical Computer Science - Algebra and coalgebra in computer science*.

[Kelly, 1982] Kelly, G. M. (1982). Basic concepts of enriched category theory,. In *London Mathematical Society Lecture Notes*, volume 64. Cambridge University Press.

[Lack, 2004] Lack, S. (2004). Composing props. *Theory and Application of Categories*, 13(9):147–163.

[Lack, 2007] Lack, S. (2007). A 2-categories companion.

[Leinster, 2003] Leinster, T. (2003). *Higher Operads, Higher Categories*. Cambridge University Press.

[Mac Lane, 1998] Mac Lane, S. (1998). *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer.

[Markl et al., 2007] Markl, M., Shnider, S., and Stasheff, J. (2007). *Operads in Algebra, Topology and Physics*. Mathematical Surveys and Monographs. American Mathematical Society.

[Marmolejo, 1997] Marmolejo, F. (1997). Doctrines whose structure forms a fully faithful adjoint string. *Theory Appl. Categ*, 3:24–44.

[Marmolejo, 1999] Marmolejo, F. (1999). Distributive laws for pseudomonads. *Theory and Application of Categories*, 5(5):91–147.

[Marmolejo and Wood, 2008] Marmolejo, F. and Wood, R. J. (2008). Coherence for pseudodistributive laws revisited. *Theory and Applications of Categories*, 20(6):74–84.

[May, 1972] May, J. (1972). *The Geometry of Iterated Loop Spaces*.

[McCrudden, 2000] McCrudden, P. (2000). Balanced coalgebroids. *Theory and Application of Categories*, 7(6):71–147.

[Rosebrugh and Wood, 2002] Rosebrugh, R. and Wood, R. J. (2002). Distributive laws and factorization. *Journal of Pure and Applied Algebra*, 175.

[Schreiber, 2011] Schreiber, U. (2011). nlab article: Equivalence of 2-categories.

[Street, 2003] Street, R. (2003). Functorial calculus in monoidal bicategories,. *Applied Categorical Structures*, 11:219–227.

[Tanaka, 2005] Tanaka, M. (2005). *Pseudo-Distributive Laws and a Unified Framework for Variable Binding*. PhD thesis, University of Edinburgh.