The Formal Theory of Monads, Univalently

Niels van der Weide
Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands

Abstract

We develop the formal theory of monads, as established by Street, in univalent foundations. This allows us to formally reason about various kinds of monads on the right level of abstraction. In particular, we define the bicategory of monads internal to a bicategory, and prove that it is univalent. We also define Eilenberg-Moore objects, and we show that both Eilenberg-Moore categories and Kleisli categories give rise to Eilenberg-Moore objects. Finally, we relate monads and adjunctions in arbitrary bicategories. Our work is formalized in Coq using the `UniMath` library.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Type theory

Keywords and phrases bicategory theory, univalent foundations, formalization, monads, Coq

Digital Object Identifier 10.4230/LIPIcs.FSCD.2023.6

Related Version *Preprint*: `https://arxiv.org/abs/2212.08515`

Supplementary Material *Software*: `https://doi.org/10.5281/zenodo.7848572` [53]
archived at `swh:1:dir:b2edfe385dd6437e5efc4fdc18c9bd48e006a144`

Funding This research was supported by the NWO project "The Power of Equality" OCENW.M20.380, which is financed by the Dutch Research Council (NWO).

Acknowledgements The author thanks Benedikt Ahrens, Deivid Vale, Herman Geuvers, and Kobe Wullaert for their comments on earlier versions of this paper. The author also thanks the Coq developers for providing the Coq proof assistant and their continuous support to keep `UniMath` compatible with Coq.

1 Introduction

Monads are ubiquitous in both mathematics and computer science, and many different kinds of monads have been considered in various settings. In functional programming, monads are used to capture computational effects [22]. Strong monads have been used to provide semantics of programming languages such as Moggi's computational λ-calculus [35, 36] and models of call-by-push-value [30]. Monads are also used in algebra to represent algebraic theories, and in fact, the class of algebraic theories is equivalent to a class of monads [20]. This result has been adapted to the enriched case as well in order to relate various notions of computation with enriched monads [16, 39, 40, 42]. Comonads, the dual notion of monads, found applications in the semantics of linear logic [11, 34].

A general setting in which all these different variations of monads can be studied, has been developed by Street [47]. This setting, known as *the formal theory of monads*, uses the fact that the notion of monad can be defined internal to an arbitrary bicategory [10], including 2-categories (which were used by Street). Each of the aforementioned kinds of monads is actually an instance of this more general notion. For example, monads in the bicategory of symmetric monoidal categories are symmetric monoidal monads, and strong monads are monads in the bicategory of so-called left actegories [13]. Comonads in a bicategory B are the same as monads in B^{co}, which is B with the 2-cells reversed. Even several kinds of

Table 1 Various notions of monads.

Bicategory	Notion of monad
Symmetric monoidal categories	Symmetric monoidal monad
Actegories [13]	Strong monad
Enriched categories [25]	Enriched monad
Bicategory of monads	Distributive law

distributive laws, including mixed distributive laws [12, 44] and iterated distributive laws [14], are instances of this notion of monad. An overview of the different kinds of monads internal to various bicategories can be found in Table 1. As such, the formal theory of monads provides a general setting to study various kinds of monads.

Foundations. In this paper, we work in **univalent foundations** [50]. Univalent foundations is an extension of intensional type theory with the univalence axiom. Roughly speaking, this axiom says that equivalent types are equal. More precisely, we have a map $\mathsf{idtoequiv}$ that sends identities $X = Y$ to equivalences from X to Y, and the univalence axiom states that $\mathsf{idtoequiv}$ is an equivalence itself. This axiom has numerous effects on the mathematics in this foundation. One consequence is function extensionality and another is that identity types must necessarily be proof relevant: since there could be multiple equivalence between two types, we could have different proofs of their equality.

In addition, the notion of category studied in this setting is that of **univalent categories** [2]. In every category C, we have a map $\mathsf{idtoiso}_{x,y} : x = y \to x \cong y$, and C is **univalent** if $\mathsf{idtoiso}_{x,y}$ is an equivalence for all $x, y : \mathsf{C}$. From the univalence axiom, one can deduce that the category of sets is univalent. The reason why this notion is interesting, is because in the set-theoretical semantics [23], univalent categories correspond to ordinary categories. Furthermore, every property expressible in type theory about univalent categories is closed under equivalence.

However, there are some challenges when working with univalent categories. For instance, the usual definition of the *Kleisli category* [31] does not give rise to a univalent category, so this category does not actually define a Kleisli category in univalent foundations. A solution to this problem has already been given: we need to use its Rezk completion [6]. However, the necessary theorems about the Kleisli category (e.g., every monad gives rise to an adjunction via the Kleisli category) have not been proven in that work.

In the present paper, we study and formalize the formal theory of monads by Street [47] in univalent foundations. More specifically, we formalize the key notions, which are the bicategory of monads and Eilenberg-Moore objects, and we illustrate them with numerous examples. We also prove the two main theorems that relate monads to adjunctions: every adjunction gives rise to a monad and in a bicategory with Eilenberg-Moore objects, every monad gives rise to an adjunction. In addition, we instantiate the formal theory of monads to deduce the main theorems about Kleisli categories. The contributions of this paper is the development of the formal theory of monads in univalent foundations and a proof that in univalent foundations, every monad gives rise to an adjunction via the Kleisli category.

The abstract setting provided by the formal theory of monads is beneficial for formalization, which is our main motivation for this work. The main theorems are only proven once in this setting, and afterwards they are instantiated to the relevant cases of interest without the need of reproving anything. In addition, we think that this work would be useful to formalize the categorical semantics of linear logic [34] or the enriched effect calculus [16].

Formalization. The results in this paper are formalized using the Coq proof assistant [48], and they are integrated in the `UniMath` library [53]. `UniMath` is under constant development, and the paper refer to the version with `git` hash 2f79746. The formalization consists of around 12,000 lines of code. More specifically, the tool `coqwc` gives the following count:

```
spec    proof comments
4470     7839       182 total
```

The main difficulty arises from the coherences that have to be proven in this development. Since we use bicategories rather than 2-categories, every coherence becomes more complicated. There also are points where univalence helps with obtaining simple and elegant proofs, such as Propositions 3.7 and 5.5. Displayed bicategories play a fundamental role in Section 3, where they are used to give a modular proof that the bicategory of monads is univalent.

Definitions, theorems, constructions, and examples in this paper are accompanied with a link that points to the relevant definition in the formalization. For example, `bicat` refers to the definition of a bicategory.

Related work. The formal theory of monads was originally developed by Street [47], and later extended by Lack and Street [28]. There are two differences between our work and Street's work. First of all, Street used strict 2-categories while we use bicategories. As has already been noticed by Lack [26], this difference is rather minor. The resulting definitions are similar: the only difference is that associators and unitors have to be put on the right places. More fundamental is the second difference: we work in univalent foundations and univalent (bi)categories whereas Street works in set-theoretic foundations. This affects the development in several ways. While both bicategories and strict 2-categories have been defined and studied in a univalent setting [1], a coherence theorem [32, 41] has not been proven in this setting. In addition, since we work in an intensional setting, working with a strict 2-category is not significantly more convenient than working with a bicategory. The reason for that, is that equality proofs of associativity and unitality are present in terms to guarantee that the whole expression is well-typed. As such, a coherence theorem would only have limited usability in our setting compared to a classical one. Another difference is that in our framework, the usual definition of the Kleisli category does not give rise to a univalent category, and we need to work with its Rezk completion instead. An overview of the main notions in bicategory theory can be found in various sources [10, 21, 27, 29].

Several formalizations have results about bicategory theory. The coherence theorem [29] is formalized in both Isabelle [37, 46] and Lean [33], but neither of those are based on univalent foundations. Some notions in bicategory theory have been formalized in Agda [38], namely in the 1Lab [49] and the Agda-categories library [19]. However, neither of these cover the formal theory of monads. We use `UniMath` [53] and its formalization of bicategories [1, 51]. Formalizations on category theory are more plentiful, and an overview can be found in [19]. Within the framework of univalent foundations, there is the HoTT library [9, 18], Agda-UniMath [45], and Cubical Agda [52]. Ahrens, Matthes, and Mörtberg formalized monads of categories in `UniMath` [53]. They also defined a notion of signature, that allows for binding, and they showed that every signature gives rise to a monad [4, 5].

Overview. We start this paper by recalling some preliminary notions in Section 2. Next we construct in Section 3 the bicategory of monads internal to bicategories and we prove that it is univalent. We illustrate the material of Section 3 with various examples in Section 4. In Section 5 we discuss Eilenberg-Moore objects. We follow that up in Section 6 by using Kleisli categories to construct Eilenberg-Moore objects in the opposite bicategory. In Sections 7 and 8 we prove some theorems in this setting. We prove in Section 7 that every adjunction gives rise to a monad and that every monad gives rise to an adjunction under mild assumptions. In Section 8 we define the notion of monadic adjunctions in an arbitrary bicategory and we characterize those using monadic adjunctions in categories. We conclude in Section 9.

2 Preliminaries

In this section, we briefly recall some of the basic notions needed in this paper. First of all, we use the notions of *propositions* and *sets* from univalent foundations. Types A for which we have $x = y$ for all $x, y : A$, are called **propositions**, and types A for which every $x = y$ is a proposition, are called **sets**. In addition, we assume that our foundation supports the **propositional truncation**: the truncation $\|A\|$ is A with all its elements identified. More concretely, we have a map $A \to \|A\|$ and for all $x, y : \|A\|$, we have $x = y$. Next we discuss some notions from bicategory theory [1, 10, 29], and we start with *bicategories*.

▶ **Definition 2.1** (`bicat`). A **bicategory** B consists of a type B of objects, for every $x, y : \mathsf{B}$ a type $x \to y$ of 1-cells, and for every $f, g : x \to y$, a *set* $f \Rightarrow g$. On this data, we have the following operations:

- For every $x : \mathsf{B}$, a 1-cell $\mathrm{id}_x : x \to x$;
- For all 1-cells $f : x \to y$ and $g : y \to z$, a 1-cell $f \cdot g : x \to z$;
- For every 1-cell $f : x \to y$, a 2-cell $\mathrm{id}_f : f \Rightarrow f$;
- For all 2-cells $\theta : f \Rightarrow g$ and $\tau : g \Rightarrow h$, a 2-cell $\theta \bullet \tau : f \Rightarrow h$;
- For every 1-cell $f : x \to y$, invertible 2-cells $\lambda_f : \mathrm{id}_x \cdot f \Rightarrow f$ and $\rho_f : f \cdot \mathrm{id}_y \Rightarrow f$;
- For all 1-cells $f : w \to x$, $g : x \to y$, and $h : y \to z$, an invertible 2-cell $\alpha_{f,g,h} : (f \cdot g) \cdot h \Rightarrow f \cdot (g \cdot h)$.

If the relevant 1-cells are clear from the context, we write λ, ρ, and α instead of λ_f, ρ_f, and $\alpha_{f,g,h}$ respectively. We can also whisker 2-cells with 1-cells in two ways. Given 1-cells and 2-cells as depicted in the diagram on the left below, we have a 2-cell $f \triangleleft \tau : f \cdot g_1 \Rightarrow f \cdot g_2$, and from 1-cells and 2-cells as depicted in the diagram on the right below, we get a 2-cell $\tau \triangleright g : f_1 \cdot g \Rightarrow f_2 \cdot g$.

$$\begin{tikzcd} x \arrow[r, "f"'] & y \arrow[r, shift left, "g_1"] \arrow[r, shift right, "g_2"'] \arrow[r, phantom, "\Downarrow\tau"] & z \end{tikzcd} \qquad \begin{tikzcd} x \arrow[r, shift left, "f_1"] \arrow[r, shift right, "f_2"'] \arrow[r, phantom, "\Downarrow\tau"] & y \arrow[r, "g"] & z \end{tikzcd}$$

The laws that need to be satisfied, can be found in the literature [1, Definition 2.1].

Note that we use diagrammatic order for composition instead of compositional order. We use the notation $\underline{\mathsf{B}(x, y)}$ for the category whose objects are 1-cells $f : x \to y$ and whose morphisms from $f : x \to y$ to $g : x \to y$ are 2-cells $\tau : f \Rightarrow g$. Given a 1-cell $f : x \to y$ and an object $w : \mathsf{B}$, we have a functor $(- \cdot f)_w : \underline{\mathsf{B}(w, x)} \to \underline{\mathsf{B}(w, y)}$, which sends a 1-cell $g : w \to x$ to $g \cdot f$ and a 2-cell $\tau : g_1 \Rightarrow g_2$ to $\tau \triangleright f$.

The core example of a bicategory in this paper is $\mathsf{UnivCat}$. Its objects are *univalent* categories, the 1-cells are functors, and the 2-cells are natural transformations. We also have a bicategory Cat whose objects are (not necessarily univalent) categories, 1-cells are functors, and 2-cells are natural transformations.

In this paper, we also make use of *univalent* bicategories. To define this property, we use that between every two objects $x, y : \mathsf{B}$, we have a type $x \simeq y$ of *adjoint equivalences* between them. In addition, for all 1-cells $f, g : x \to y$, there is a type $f \cong g$ of *invertible 2-cells* between them. For the precise definition of these notions, we refer the reader to the literature [1, Definitions 2.4 and 2.5].

▶ **Definition 2.2.** Let B be a bicategory

- (`is_univalent_2_1`) Using path induction, we define a function $\mathsf{idtoiso}^{2,1}_{f,g} : f = g \to f \cong g$ for all 1-cells $f, g : x \to y$. A bicategory is **locally univalent** if $\mathsf{idtoiso}^{2,1}_{f,g}$ is an equivalence for all f and g.
- (`is_univalent_2_0`) Using path induction, we define a function $\mathsf{idtoiso}^{2,0}_{x,y} : x = y \to x \simeq y$ for all objects x and y. We say that B is **globally univalent** if $\mathsf{idtoiso}^{2,0}_{x,y}$ is an equivalence for all x and y.
- (`is_univalent_2`) A bicategory is **univalent** if it is both locally and globally univalent.

The bicategory $\mathsf{UnivCat}$ of univalent categories is both locally and globally univalent. However, Cat, whose objects objects are not required to be univalent, is neither.

If we have a bicategory B, then we define the bicategory B^{op} by 'reversing the 1-cells' in B. More precisely, objects are the same as objects in B, 1-cells from x to y in B^{op} are 1-cells $y \to x$ in B, while 2-cells from $f : y \to x$ to $g : y \to x$ are 2-cells $f \Rightarrow g$ in B. In addition, from a bicategory B, we obtain B^{co} by 'reversing the 2-cells'. Objects and 1-cells in B^{co} are the same as objects and 1-cells in B respectively, but a 2-cell from f to g in B^{co} is a 2-cell $g \Rightarrow f$ in B. Next we define *pseudofunctors*.

▶ **Definition 2.3** (`psfunctor`)**.** Let B_1 and B_2 be bicategories. A **pseudofunctor** $F : \mathsf{B}_1 \to \mathsf{B}_2$ consists of

- A function $F : \mathsf{B}_1 \to \mathsf{B}_2$;
- For every $x, y : \mathsf{B}_1$ a function that maps $f : x \to y$ to $F\,f : F\,x \to F\,y$;
- For all 1-cell $f, g : x \to y$ a function that maps $\theta : f \Rightarrow g$ to $F\,\theta : F\,f \Rightarrow F\,g$;
- For every $x : \mathsf{B}_1$, an invertible 2-cell $F_i(x) : \mathrm{id}_{F\,x} \Rightarrow F(\mathrm{id}_x)$;
- For all $f : x \to y$ and $g : y \to z$, an invertible 2-cell $F_c(f, g) : F(f) \cdot F(g) \Rightarrow F(f \cdot g)$.

The coherences that need to be satisfied, can be found in the literature [1, Definition 2.12].

In applications, we are interested in a wide variety of bicategories beside $\mathsf{UnivCat}$, and among those are $\mathsf{UnivCat}_{\mathsf{Terminal}}$ and $\mathsf{SymMonUnivCat}$. These examples have something in common: their objects are categories equipped with some extra structure, the 1-cells are structure preserving functors, while the 2-cells are structure preserving natural transformations. We capture this pattern using *displayed bicategories*. This notion is an adaptation of *displayed categories* to the bicategorical setting [3].

To get an idea of what displayed bicategories are, let us first briefly discuss displayed categories. A displayed category D over C represents structure and properties to be added to the objects and morphisms of C. For every object $x : \mathsf{C}$, we have a type of *displayed objects* D_x and for every morphism $f : x \to y$ and displayed objects $\overline{x} : \mathsf{D}_x$ and $\overline{y} : \mathsf{D}_y$, we have a set $\overline{x} \xrightarrow{f} \overline{y}$ of *displayed morphisms*. For example, if for C we take the category of sets, an example of a displayed category would be group structures. The displayed objects over a set X are group structures over X, while the displayed morphisms over $f : X \to Y$ between two group structures are proofs that f preserves the group operations. Every displayed category D gives rise to the *total category* $\int \mathsf{D}$ and a functor $\pi_{\mathsf{D}} : \int \mathsf{D} \to \mathsf{C}$. In the example we mentioned before, $\int \mathsf{D}$ would be the category of groups and π_{D} maps a group to its underlying set.

In the bicategorical setting, we use a similar approach, but a displayed bicategory should not only have displayed objects and 1-cells, but also displayed 2-cells. More precisely, we define displayed bicategories as follows.

▶ **Definition 2.4** (`disp_bicat`). A **displayed bicategory** D over a bicategory B consists of

- For every $x : \mathsf{B}$ a type D_x of **displayed objects**;
- For all 1-cells $f : x \to y$ and displayed objects $\overline{x} : D_x$ and $\overline{y} : D_y$, a type $\overline{x} \xrightarrow{f} \overline{y}$ of **displayed morphisms**;
- For all 2-cells $\theta : f \Rightarrow g$ and displayed morphisms $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$ and $\overline{g} : \overline{x} \xrightarrow{g} \overline{y}$ a set $\overline{f} \overset{\theta}{\Longrightarrow} \overline{g}$ of **displayed 2-cells**.

In addition, there are displayed versions of every operation of bicategories. For example, for every $x : \mathsf{B}$ and $\overline{x} : D_x$, we have the displayed identity $\overline{\mathrm{id}_x} : \overline{x} \xrightarrow{\mathrm{id}_x} \overline{x}$, and from two displayed morphisms $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$ and $\overline{g} : \overline{y} \xrightarrow{g} \overline{z}$ we get a displayed morphism $\overline{f} \cdot \overline{g} : \overline{x} \xrightarrow{f \cdot g} \overline{z}$. A list of the operations and laws can be found in the literature [1, Definition 6.1].

Just like for displayed categories, every displayed bicategory D gives rise to bicategory $\int \mathsf{D}$ and a pseudofunctor $\int \mathsf{D} \to \mathsf{B}$.

▶ **Problem 2.5.** *Given a displayed bicategory* D *over* B*, to construct a bicategory* $\int \mathsf{D}$ *and a pseudofunctor* $\pi_{\mathsf{D}} : \int \mathsf{D} \to \mathsf{B}$.

▶ **Construction 2.6** (for Problem 2.5; `total_bicat`). The bicategory $\int \mathsf{D}$ is defined as follows:

- Its objects are pairs $x : \mathsf{B}$ together with $\overline{x} : \mathsf{D}_x$;
- Its 1-cells from $(x, \overline{x})$ to $(y, \overline{y})$ are pairs $f : x \to y$ together with $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$;
- Its 2-cells from $(f, \overline{f})$ to $(g, \overline{g})$ are pairs $\tau : f \Rightarrow g$ together with $\overline{\tau} : \overline{f} \overset{\tau}{\Longrightarrow} \overline{g}$.

The pseudofunctor π_{D} sends objects $(x, \overline{x})$ to x, 1-cells $(f, \overline{f})$ to f, and 2-cells $(\theta, \overline{\theta})$ to θ. ⌟

Univalent displayed bicategories are defined similarly to univalent bicategories, and the precise definition can be found elsewhere [1, Definition 7.3]. If we have a displayed bicategory D over B and if both B and D are univalent, then $\int \mathsf{D}$ is univalent as well. This gives a modular way to construct univalent bicategories.

In numerous different examples, we look at displayed bicategories whose displayed 2-cells are actually trivial, in a certain sense. More precisely, we look at two properties. One of them expresses that between all displayed 1-cells $\overline{f} : \overline{x} \xrightarrow{f} \overline{y}$ and $\overline{g} : \overline{x} \xrightarrow{g} \overline{y}$, there is at most one displayed 2-cell. A displayed bicategory satisfying hat property, is called a *local preorder*: it expresses that the type $\overline{f} \overset{\tau}{\Longrightarrow} \overline{g}$ is a proposition. The other property is *locally groupoidal*, and it says that every displayed 2-cell over an invertible 2-cell is again invertible.

▶ **Definition 2.7** (`disp_cell_unit_bicat`). Suppose, we have a bicategory B and

- For each $x : \mathsf{B}$ a type D_x;
- For every 1-cell $f : x \to y$ and elements $\overline{x} : \mathsf{D}_x$ and $\overline{y} : \mathsf{D}_y$ a type $\mathsf{D}_{\overline{x},\overline{y},f}$;
- For every $x : \mathsf{B}$ and $\overline{x} : \mathsf{D}_x$ an inhabitant of $\mathsf{D}_{\overline{x},\overline{x},\mathrm{id}_x}$;
- For all $\overline{f} : \mathsf{D}_{\overline{x},\overline{y},f}$ and $\overline{g} : \mathsf{D}_{\overline{y},\overline{z},g}$ an inhabitant of $\mathsf{D}_{\overline{x},\overline{z},f \cdot g}$.

Then we get a displayed bicategory over B whose 2-cells are inhabitants of the unit type.

Note that every displayed bicategory constructed using Definition 2.7 is both a local preorder and locally groupoidal. To illustrate the notion of displayed category, let us look at some examples. These were already considered in previous work [7, 54].

▶ **Example 2.8** (`univ_cat_with_terminal_obj`). Using Definition 2.7, we define a displayed bicategory $\mathsf{dUnivCat}_{\mathsf{Terminal}}$ over $\mathsf{UnivCat}$.

- Objects over C are terminal objects T_{C} in C;
- Displayed 1-cells over $F : \mathsf{C}_1 \to \mathsf{C}_2$ are proofs that F preserves terminal objects.

We define $\mathsf{UnivCat}_{\mathsf{Terminal}}$ to be $\int \mathsf{dUnivCat}_{\mathsf{Terminal}}$.

▶ **Example 2.9** (`disp_bicat_univmon`)**.** We define a displayed bicategory $\mathsf{dMonUnivCat}$ over $\mathsf{UnivCat}$ as follows:

- Objects over C are monoidal structures over C;
- 1-cells over $F : \mathsf{C}_1 \to \mathsf{C}_2$ are structures that F is a lax monoidal functor;
- 2-cells over $\theta : F \Rightarrow G$ are proofs that θ is a monoidal transformation.

We denote its total bicategory by $\mathsf{MonUnivCat}$.

Note that similarly, one can define a displayed bicategory $\mathsf{dSymMonUnivCat}$ over $\mathsf{UnivCat}$ whose total bicategory $\mathsf{SymMonUnivCat}$ is the bicategory of symmetric monoidal categories. In the remainder of this paper, we use several operations on displayed bicategories.

▶ **Definition 2.10.** We have the following operations on displayed bicategories:

1. (`disp_dirprod_bicat`) Given displayed bicategories D_1 and D_2 over B, we construct a displayed bicategory $\mathsf{D}_1 \times \mathsf{D}_2$ over B whose objects, 1-cells and 2-cells are pairs of objects, 1-cells, and 2-cells in D_1 and D_2 respectively.
2. (`sigma_bicat`) Suppose that we have a displayed bicategory D_1 over B and a displayed bicategory D_2 over $\int \mathsf{D}_1$. We construct a displayed bicategory $\sum(\mathsf{D}_2)$ over B by setting the objects over x to be pairs of $(\overline{x}, \overline{\overline{x}})$ of objects $\overline{x} : D_{1x}$ and $\overline{\overline{x}} : \mathsf{D}_{2(x,\overline{x})}$.
3. (`disp_fullsubbicat`) Let B be a bicategory and let P be a predicate on the objects of B. We define a displayed bicategory $\mathsf{dFullSub}(P)$ over B by setting the displayed objects over x to be proofs of $P(x)$ and the 1-cells and 2-cells are inhabitants of the unit type.

The last notion we discuss, is the notion of a *section* of a displayed bicategory. Intuitively, a section assigns to every x a displayed object $\overline{x} : \mathsf{D}_x$ in a pseudofunctorial way. As such every section s of D induces a pseudofunctor $\mathsf{Section}(s) : \mathsf{B} \to \int D$ such that every $x : \mathsf{B}$ is *definitionally* equal to $\pi_{\mathsf{D}}(\mathsf{Section}(s)(x))$.

▶ **Definition 2.11** (`section_disp_bicat`)**.** A **section** s of D consists of

- For every object $x : \mathsf{B}$ a displayed object $s_x : D_x$;
- For every 1-cell $f : x \to y$ a displayed 1-cell $s_f : s_x \xrightarrow{f} s_y$;
- For every 2-cell $\theta : f \Rightarrow g$ a displayed 2-cell $s_\theta : s_f \overset{\theta}{\Rightarrow} s_g$;
- For every $x : \mathsf{B}$ an invertible 2-cell $s_i(x) : \overline{\mathrm{id}_{s_x}} \xRightarrow{\mathrm{id}_{\mathrm{id}_x}} s_{\mathrm{id}_x}$;
- For every $f : x \to y$ and $g : y \to z$ an invertible 2-cell $s_c(f,g) : s_f \cdot s_g \xRightarrow{\mathrm{id}_{f \cdot g}} s_{f \cdot g}$.

We also require several coherences reminiscent of the laws of pseudofunctors, and for a precise description of those, we refer the reader to the formalization.

▶ **Problem 2.12.** *Given a displayed bicategory* D *on* B *and a section* s *on* D*, to construct a pseudofunctor* $\mathsf{Section}(s) : \mathsf{B} \to \int D$*.*

▶ **Construction 2.13** (for Problem 2.12; `section_to_psfunctor`)**.** We define the pseudofunctor $\mathsf{Section}(s)$ as follows: it sends objects x to the pair (x, s_x), 1-cells f to (f, s_f), and 2-cells $\tau : f \Rightarrow g$ to (τ, s_τ). ⌟

3 The Bicategory of Monads

Two concepts play a key role in the formal theory of monads: the bicategory of monads $\mathsf{Mnd}(\mathsf{B})$ internal to B and Eilenberg-Moore objects. In this section, we study the first concept, and we use displayed bicategories to construct the bicategory $\mathsf{Mnd}(\mathsf{B})$ given a bicategory B. We also show that $\mathsf{Mnd}(\mathsf{B})$ must be univalent if B is.

The main idea behind the construction is to split up monads in several independent parts. We first define $\mathsf{dEndo}(\mathsf{B})$ whose displayed objects over x are 1-cells $e : x \to x$. After that, we define $\mathsf{dUnit}(\mathsf{B})$ and $\mathsf{dMult}(\mathsf{B})$ whose displayed objects are the unit and multiplication of the monad respectively. We finally take a full subcategory for the monad laws.

▶ **Definition 3.1** (`disp_end`)**.** Let B be a bicategory. Define a displayed bicategory $\mathsf{dEndo}(\mathsf{B})$ over B as follows:

- The displayed objects over x are 1-cells $e_x : x \to x$;
- The displayed 1-cells over $f : x \to y$ from $e_x : x \to x$ to $e_y : y \to y$ are 2-cells τ_f

$$\begin{tikzcd} x \arrow[r, "e_x"] \arrow[d, "f"'] \arrow[dr, phantom, "\Uparrow\tau_f"] & x \arrow[d, "f"] \\ y \arrow[r, "e_y"'] & y \end{tikzcd}$$

- The displayed 2-cells over $\tau : f \Rightarrow g$ from $\theta_f : f \cdot e_y \Rightarrow e_x \cdot f$ to $\theta_g : g \cdot e_y \Rightarrow e_x \cdot g$ are proofs that the following diagram commutes

$$\begin{tikzcd} f \cdot e_y \arrow[r, Rightarrow, "\theta_f"] \arrow[d, Rightarrow, "\tau \triangleright e_y"'] & e_x \cdot f \arrow[d, Rightarrow, "e_x \triangleleft \tau"] \\ g \cdot e_y \arrow[r, Rightarrow, "\theta_g"'] & e_x \cdot g \end{tikzcd}$$

We define $\mathsf{Endo}(B)$ by $\int \mathsf{dEndo}(\mathsf{B})$.

Next we define two displayed bicategories over $\mathsf{Endo}(\mathsf{B})$. For both, we use Definition 2.7.

▶ **Definition 3.2** (`disp_add_unit`)**.** Given a bicategory B, we define the displayed bicategory $\mathsf{dUnit}(\mathsf{B})$ over $\mathsf{Endo}(\mathsf{B})$ as follows

- The displayed objects over (x, e_x) are 2-cells $\eta_x : \mathrm{id}_x \Rightarrow e_x$;
- The displayed 1-cells over (f, θ_f) where $f : x \to y$ and $\theta_f : f \cdot e_y \Rightarrow e_x \cdot f$ from $\eta_x : \mathrm{id}_x \Rightarrow e_x$ to $\eta_y : \mathrm{id}_y \Rightarrow e_y$ are proofs that the following diagram commutes

$$\begin{tikzcd} f \arrow[rr, Rightarrow, "\lambda^{-1}"] \arrow[d, Rightarrow, "\rho^{-1}"'] & & \mathrm{id}_x \cdot f \arrow[d, Rightarrow, "\eta_x \triangleright f"] \\ f \cdot \mathrm{id}_y \arrow[r, Rightarrow, "f \triangleleft \eta_y"'] & f \cdot e_y \arrow[r, Rightarrow, "\theta_f"'] & e_x \cdot f \end{tikzcd}$$

▶ **Definition 3.3** (`disp_add_mu`)**.** Given a bicategory B, we define the displayed bicategory $\mathsf{dMult}(\mathsf{B})$ over $\mathsf{Endo}(\mathsf{B})$ as follows

- The displayed objects over (x, e_x) are 2-cells $\mu_x : e_x \cdot e_x \Rightarrow e_x$;
- The displayed 1-cells over (f, θ_f) where $f : x \to y$ and $\theta_f : f \cdot e_y \Rightarrow e_x \cdot f$ from $\mu_x : e_x \cdot e_x \Rightarrow e_x$ to $\mu_y : e_y \cdot e_y \Rightarrow e_y$ are proofs that the following diagram commutes

$$\begin{tikzcd} f \cdot (e_y \cdot e_y) \arrow[rr, Rightarrow, "f \triangleleft \mu_y"] \arrow[d, Rightarrow, "\alpha"'] & & f \cdot e_y \arrow[rr, Rightarrow, "\theta_f"] & & e_x \cdot f \\ (f \cdot e_y) \cdot e_y \arrow[r, Rightarrow, "\theta_f \triangleright e_y"'] & (e_x \cdot f) \cdot e_y \arrow[r, Rightarrow, "\alpha^{-1}"'] & e_x \cdot (f \cdot e_y) \arrow[r, Rightarrow, "e_x \triangleleft \theta_f"'] & e_x \cdot (e_x \cdot f) \arrow[r, Rightarrow, "\alpha"'] & (e_x \cdot e_x) \cdot f \arrow[u, Rightarrow, "\mu_x \triangleright f"'] \end{tikzcd}$$

Next we define $\mathsf{dMndData}(\mathsf{B})$ to be $\sum(\mathsf{dUnit}(\mathsf{B}) \times \mathsf{dMult}(\mathsf{B}))$ and we denote its total bicategory by $\mathsf{MndData}(\mathsf{B})$. To obtain $\mathsf{Mnd}(\mathsf{B})$, we take a full subbicategory.

▶ **Definition 3.4** (disp_mnd)**.** For a bicategory B, we define the predicate isMnd over $\mathsf{MndData}(\mathsf{B})$: given $(x, (e_x, (\eta_x, \mu_x)))$, the following diagrams commute

$$\begin{tikzcd}
e_x \arrow[r, Rightarrow, "\rho^{-1}"] \arrow[drr, Rightarrow, "\mathrm{id}_x"'] & e_x \cdot \mathrm{id}_x \arrow[r, Rightarrow, "e_x \triangleleft \eta_x"] & e_x \cdot e_x \arrow[d, Rightarrow, "\mu_x"] & \mathrm{id}_x \cdot e_x \arrow[l, Rightarrow, "\eta_x \triangleright e_x"'] & e_x \arrow[l, Rightarrow, "\lambda^{-1}"'] \arrow[dll, Rightarrow, "\mathrm{id}_{e_x}"] \\
 & & e_x & &
\end{tikzcd}$$

$$\begin{tikzcd}
e_x \cdot (e_x \cdot e_x) \arrow[rr, Rightarrow, "e_x \triangleleft \mu_x"] \arrow[d, Rightarrow, "\alpha"'] & & e_x \cdot e_x \arrow[d, Rightarrow, "\mu_x"] \\
(e_x \cdot e_x) \cdot e_x \arrow[r, Rightarrow, "\mu_x \triangleright e_x"'] & e_x \cdot e_x \arrow[r, Rightarrow, "\mu_x"'] & e_x
\end{tikzcd}$$

We define $\mathsf{dMnd}(\mathsf{B})$ to be $\sum(\mathsf{dFullSub}(\mathsf{isMnd}))$, and we denote its total bicategory by $\mathsf{Mnd}(\mathsf{B})$.

Note that the predicate isMnd contains all monads laws. Alternatively, we could have defined a displayed bicategory for each monad law. We refrained from doing so, because that would not further simplify the proof that $\mathsf{Mnd}(\mathsf{B})$ is univalent.

Before we continue, let us discuss the objects, 1-cells, and 2-cells of $\mathsf{Mnd}(\mathsf{B})$, and fix the relevant notation for the remainder of this paper. This also explains what the displayed objects, 1-cells, and 2-cells in $\mathsf{dMnd}(\mathsf{B})$ are. The data of a monad m in B consists of

- an object $\mathsf{ob}_m : \mathsf{B}$;
- a 1-cell $\mathsf{mor}_m : \mathsf{ob}_m \to \mathsf{ob}_m$;
- a 2-cell $\eta_m : \mathrm{id}_{\mathsf{ob}_m} \Rightarrow m$;
- a 2-cell $\mu_m : m \cdot m \Rightarrow m$.

If no confusion arises, we write m instead of mor_m.

The data of a monad morphism $f : m_1 \to m_2$ between monads m_1 and m_2 consists of a 1-cell $\mathsf{mor}_f : \mathsf{ob}_{m_1} \to \mathsf{ob}_{m_2}$ and a 2-cell $\mathsf{cell}_f : f \cdot m_2 \Rightarrow m_1 \cdot f$. Lastly, the data of monad 2-cell $\gamma : f_1 \Rightarrow f_2$ between monad morphisms is a 2-cell $\mathsf{cell}_\gamma : f_1 \Rightarrow f_2$. We write f instead of mor_f and γ instead of cell_γ if no confusion arises.

Next we look at the univalence of $\mathsf{Mnd}(\mathsf{B})$, and to prove it, we use displayed univalence [1, Definition 7.3]. This way, it suffices to prove the displayed univalence of $\mathsf{dEndo}(\mathsf{B})$, $\mathsf{dUnit}(\mathsf{B})$, and $\mathsf{dMult}(\mathsf{B})$. We also need B to be univalent.

▶ **Proposition 3.5** (is_univalent_2_mnd)**.** *If B is univalent, then so is $\mathsf{Mnd}(\mathsf{B})$.*

We also characterize invertible 2-cells and adjoint equivalences in $\mathsf{Mnd}(\mathsf{B})$.

▶ **Proposition 3.6** (is_invertible_mnd_2cell)**.** *A 2-cell γ between monad morphisms is invertible, if the underlying 2-cell $\mathsf{cell}_\gamma : f_1 \Rightarrow f_2$ is invertible.*

▶ **Proposition 3.7** (to_equivalence_mnd)**.** *Let B be a univalent bicategory and let $f : m_1 \to m_2$ be a 1-cell in $\mathsf{Mnd}(B)$. If mor_f is an adjoint equivalence and cell_f is an invertible 2-cell, then f is an adjoint equivalence.*

Note that in Proposition 3.7, we assume that B is univalent. Because of univalence, it suffices to prove this proposition assuming that mor_f is the identity 1-cell, which simplifies the involved coherences. The same idea can be used to show that pointwise pseudonatural adjoint equivalences are adjoint equivalences in the bicategory of pseudofunctors.

4 Examples of Monads

Next we look at examples of monads, and we start by characterizing monads internal to several bicategories. Let us start by observing that monads in the bicategory $\mathsf{UnivCat}$ of categories correspond to monads as how they usually are defined in category theory. However, since this notion of monad is defined in *every* bicategory, we can also look at other bicategories, such as $\mathsf{SymMonUnivCat}$,and $\mathsf{UnivCat}_{\mathsf{Terminal}}$.

In a wide variety of applications, one is interested in monads in a bicategory of categories with some extra structure. For example, symmetric monoidal monads are monads internal to the bicategory of symmetric monoidal categories. Strong monads are monads in the bicategory of left actegories. The bicategories in these two examples can be constructed as a total bicategory of some displayed bicategory over $\mathsf{UnivCat}$. To characterize monads in total bicategories, we define *displayed monads.*

▶ **Definition 4.1** (`disp_mnd`)**.** Let B be a bicategory and let D be a displayed bicategory over B and suppose that D is a local preorder and locally groupoidal. A **displayed monad** $\overline{m}$ over a monad m in B consists of

- a displayed object $\overline{\mathsf{ob}_m} : \mathsf{D}_{\mathsf{ob}_m}$;
- a displayed 1-cell $\overline{m} : \overline{\mathsf{ob}_m} \xrightarrow{m} \overline{\mathsf{ob}_m}$;
- a displayed 1-cell $\overline{\eta_m} : \overline{\mathrm{id}_{\mathsf{ob}_m}} \xRightarrow{\eta_m} \overline{m}$;
- a displayed 1-cell $\overline{\mu_m} : \overline{m} \cdot \overline{m} \xRightarrow{\mu_m} \overline{m}$.

Note that we do not require any coherences in Definition 4.1, because the involved displayed bicategory is assumed to be a local preorder.

▶ **Problem 4.2.** *Given a monad m and a displayed monad $\overline{m}$ over m, to construct a monad $\int m$ in $\int D$.*

▶ **Construction 4.3** (for Problem 4.2; `make_mnd_total_bicat`)**.** We construct $\int m$ as follows

- We define the object $\mathsf{ob}_{\int m}$ to be $(\mathsf{ob}_m, \overline{\mathsf{ob}_m})$;
- We define the 1-cell $\int m$ to be $(m, \overline{m})$;
- We define the unit $\eta_{\int m}$ to be $(\eta_m, \overline{\eta_m})$;
- We define multiplication $\mu_{\int m}$ to be $(\mu_m, \overline{\mu_m})$. ⌟

▶ **Example 4.4** (`make_mnd_univ_cat_with_terminal_obj`)**.** Every displayed monad in $\mathsf{dUnivCat}_{\mathsf{Terminal}}$ over m consists of a terminal object in ob_m and a proof that m preserves terminal objects.

Analogously, we can characterize monads in $\mathsf{SymMonUnivCat}$. Next we look at monads in B^{op} and B^{co}.

▶ **Example 4.5.** We characterize monads in B^{op} and B^{co}.

- (`MonadsInOp1Bicat.v`) Monads in B^{op} are the same as monads in B. However, 1-cells in $\mathsf{Mnd}(\mathsf{B}^{\mathsf{op}})$ are different from 1-cells in $\mathsf{Mnd}(\mathsf{B})$. If we have a 1-cell $f : m_1 \to m_2$ in $\mathsf{Mnd}(\mathsf{B}^{\mathsf{op}})$, then the cell cell_f gives rise to a 2-cell $m_2 \cdot f \Rightarrow f \cdot m_1$. Hence, 1-cells in $\mathsf{Mnd}(\mathsf{B}^{\mathsf{op}})^{\mathsf{op}}$ are the same as oplax monad morphisms in B.
- (`MonadsInOp2Bicat.v`) Suppose, we have $m : \mathsf{Mnd}(\mathsf{B}^{\mathsf{co}})$. Then $\mathsf{ob}_m : \mathsf{B}$ and we also have a 1-cell $m : \mathsf{ob}_m \to \mathsf{ob}_m$ in B. However, since the direction of the 2-cells are reversed in B^{co}, the 2-cells η_m and μ_m give rise to a 2-cell $m \Rightarrow \mathrm{id}_{\mathsf{ob}_m}$ and $m \Rightarrow m \cdot m$ respectively. As such, monads in B^{co} are the same as comonads in B.

▶ **Example 4.6** (`mnd_mnd_to_distr_law`)**.** Objects in $m : \mathsf{Mnd}(\mathsf{Mnd}(\mathsf{B}))$ are distributive laws between monads. To see why, observe that the object ob_m is a monad in B. In addition, we can construct another monad m' in B as follows:

- The object $\mathsf{ob}_{m'}$ is $\mathsf{ob}_{\mathsf{ob}_m}$;
- The endomorphism is m, which is a 1-cell from $\mathsf{ob}_{\mathsf{ob}_m}$ to $\mathsf{ob}_{\mathsf{ob}_m}$;
- The unit and multiplication are the underlying 2-cells of η_m and μ_m respectively.

The 2-cell cell_m is the 2-cell of the distributive law, and the laws are the proofs that η_m and μ_m are 2-cells in $\mathsf{Mnd}(\mathsf{B})$.

Analogously, we can show that objects in $\mathsf{Mnd}(\mathsf{Mnd}(\mathsf{B}^{\mathsf{co}})^{\mathsf{co}})$ are *mixed distributive laws* in B [12, 44]. We can also look at *iterated distributive laws*, which are monads in $\mathsf{Mnd}^n(\mathsf{B})$ [14].

Next we give two general constructions of monads. First of all, we consider the identity monad: on every object $x : \mathsf{B}$, we construct a monad $\mathsf{idMnd}(x)$.

▶ **Problem 4.7.** *Given a bicategory* B*, to construct a section on* $\mathsf{dMnd}(\mathsf{B})$*.*

▶ **Construction 4.8** (for Problem 4.7; `mnd_section_disp_bicat`)**.** To construct the desired section, we define the identity monad $\mathsf{idMnd}(x)$ for every x.

- The object is x;
- The 1-cell is $\mathrm{id}_x : x \to x$;
- The unit is $\mathrm{id}_{\mathrm{id}_x} : \mathrm{id}_x \Rightarrow \mathrm{id}_x$;
- The multiplication is $\lambda : \mathrm{id}_x \cdot \mathrm{id}_x \Rightarrow \mathrm{id}_x$. ⌟

In the remainder, we only use the pseudofunctor arising from Construction 4.8 and Construction 2.13, and this pseudofunctor is denoted as $\mathsf{idMnd} : \mathsf{B} \to \mathsf{Mnd}(\mathsf{B})$. Second, we notice that every monad $m : \mathsf{Mnd}(\mathsf{B})$ gives rise to a monad of categories. This example is used in Section 5.

▶ **Problem 4.9.** *Given a monad m in a bicategory* B *and an object x, to construct a monad* $\mathsf{HomMnd}_x(m)$ *on* $\underline{\mathsf{B}(x, \mathsf{ob}_m)}$*.*

▶ **Construction 4.10** (for Problem 4.9; `mnd_to_cat_Monad`)**.** The monad $\mathsf{HomMnd}_x(m)$ is defined as follows:

- The endofunctor is $(- \cdot m)_x : \underline{\mathsf{B}(x, \mathsf{ob}_m)} \to \underline{\mathsf{B}(x, \mathsf{ob}_m)}$.
- For every $f : x \to \mathsf{ob}_m$, the unit is defined to be

$$f \xRightarrow{\rho^{-1}} f \cdot \mathrm{id}_{\mathsf{ob}_m} \xRightarrow{f \lhd \eta_m} f \cdot m$$

- For every $f : x \to \mathsf{ob}_m$, the multiplication is defined to be

$$(f \cdot m) \cdot m \xRightarrow{\alpha^{-1}} f \cdot (m \cdot m) \xRightarrow{f \lhd \mu_m} f \cdot m$$ ⌟

Next we show that pseudofunctors preserve monads.

▶ **Problem 4.11.** *Given bicategories* B_1 *and* B_2*, a pseudofunctor* $F : \mathsf{B}_1 \to \mathsf{B}_2$*, and a monad* $m : \mathsf{Mnd}(\mathsf{B}_1)$*, to construct a monad* $F(m) : \mathsf{Mnd}(\mathsf{B}_2)$*.*

▶ **Construction 4.12** (for Problem 4.11; `psfunctor_on_mnd`). The object of $F(m)$ is $F(\mathsf{ob}_m)$ while the 1-cell is $F(m) : F(\mathsf{ob}_m) \to F(\mathsf{ob}_m)$. The unit and multiplication are constructed using to the following pasting diagrams respectively

$$\begin{tikzcd}[column sep=huge] F(\mathsf{ob}_m) \arrow[r, "F(\mathrm{id}_{\mathsf{ob}_m})" description] \arrow[r, bend left=60, "\mathrm{id}_{F(\mathsf{ob}_m)}"] \arrow[r, bend right=60, "F(m)"'] \arrow[r, phantom, shift left=4, "\Downarrow F_i(\mathsf{ob}_m)"] \arrow[r, phantom, shift right=4, "\Downarrow F(\eta_m)"] & F(\mathsf{ob}_m) \end{tikzcd} \qquad \begin{tikzcd}[column sep=huge] F(\mathsf{ob}_m) \arrow[r, "F(m \cdot m)" description] \arrow[r, bend left=60, "F(m) \cdot F(m)"] \arrow[r, bend right=60, "F(m)"'] \arrow[r, phantom, shift left=4, "\Downarrow F_c(m,m)"] \arrow[r, phantom, shift right=4, "\Downarrow F(\mu_m)"] & F(\mathsf{ob}_m) \end{tikzcd}$$

For a proof of the monad laws, we refer the reader to the formalization. ⌟

In fact, if we have a pseudofunctor $F : \mathsf{B}_1 \to \mathsf{B}_2$, then we obtain a pseudofunctor $\mathsf{Mnd}(F) : \mathsf{Mnd}(\mathsf{B}_1) \to \mathsf{Mnd}(\mathsf{B}_2)$. Next we show that monads can be composed if we have a distributive law between them.

▶ **Example 4.13** (`compose_mnd`). Suppose that we have a distributive law τ between monads m_1 and m_2 (Example 4.6). Then we define a monad $m_1 \cdot m_2$ as follows:

- The object is ob_{m_1} (which is definitionally equal to ob_{m_2});
- The 1-cell is $m_1 \cdot m_2$;
- The unit is constructed as the following composition of 2-cells

$$\mathrm{id}_{\mathsf{ob}_{m_1}} \xRightarrow{\lambda^{-1}} \mathrm{id}_{\mathsf{ob}_{m_1}} \cdot \mathrm{id}_{\mathsf{ob}_{m_1}} \xRightarrow{\eta_{m_1} \triangleright \mathrm{id}_{\mathsf{ob}_{m_1}}} m_1 \cdot \mathrm{id}_{\mathsf{ob}_{m_1}} \xRightarrow{m_1 \triangleleft \eta_{m_2}} m_1 \cdot m_2$$

- The multiplication is the following composition of 2-cells

$$\begin{aligned} (m_1 \cdot m_2) \cdot (m_1 \cdot m_2) &\xRightarrow{\alpha^{-1}} m_1 \cdot (m_2 \cdot (m_1 \cdot m_2)) \xRightarrow{m_1 \triangleleft \alpha} m_1 \cdot ((m_2 \cdot m_1) \cdot m_2) \\ &\xRightarrow{m_1 \triangleleft (\tau \triangleright m_2)} m_1 \cdot ((m_1 \cdot m_2) \cdot m_2) \xRightarrow{m_1 \triangleleft \alpha^{-1}} m_1 \cdot (m_1 \cdot (m_2 \cdot m_2)) \xRightarrow{\alpha} (m_1 \cdot m_1) \cdot (m_2 \cdot m_2) \\ &\xRightarrow{\mu_{m_1} \triangleright (m_2 \cdot m_2)} m_1 \cdot (m_2 \cdot m_2) \xRightarrow[m_1 \triangleleft \mu_{m_2}]{} m_1 \cdot m_2 \end{aligned}$$

5 Eilenberg-Moore Objects

The second important concept in the formal theory of monads is the notion of *Eilenberg-Moore objects*. An important property of monads in category theory is that every monad gives rise to an adjunction. One can do this in two ways: either via Eilenberg-Moore categories or via Kleisli categories. In this section, we study Eilenberg-Moore objects, which formulate Eilenberg-Moore categories in bicategorical terms.

Note that the terminology in this section is slightly differently compared to what was used by Street [47]. Whereas Street would say that a bicategory admits the construction of algebras, we follow [24, 28, 43] and we say that a bicategory has Eilenberg-Moore objects. Our notions are also formulated slightly differently, because we use *Eilenberg-Moore cones*.

▶ **Definition 5.1** (`em_cone`)**.** Let B be a bicategory and let m be a monad in B. An **Eilenberg-Moore cone** for m consists of an object e together with a 1-cell $\mathsf{idMnd}(e) \to m$.

More concretely, an Eilenberg-Moore cone e for a monad m consists of

- An object ob_e;
- A 1-cell $\mathsf{mor}_e : \mathsf{ob}_e \to \mathsf{ob}_m$;
- A 2-cell $\mathsf{cell}_e : \mathsf{mor}_e \cdot m \Rightarrow \mathrm{id}_x \cdot \mathsf{mor}_e$

such that the following diagrams commutes

$$\begin{tikzcd}
\mathsf{mor}_e \arrow[r, Rightarrow, "\rho^{-1}"] \arrow[rrd, Rightarrow, "\lambda^{-1}"'] & \mathsf{mor}_e \cdot \mathrm{id}_{\mathsf{ob}_m} \arrow[r, Rightarrow, "\mathsf{mor}_e \triangleleft \eta_m"] & \mathsf{mor}_e \cdot m \arrow[d, Rightarrow, "\mathsf{cell}_e"] \\
 & & \mathrm{id}_x \cdot \mathsf{mor}_e
\end{tikzcd}$$

$$\begin{tikzcd}
\mathsf{mor}_e \cdot (m \cdot m) \arrow[rrr, Rightarrow, "\mathsf{mor}_e \triangleleft \mu_m"] \arrow[d, Rightarrow, "\alpha"'] & & & \mathsf{mor}_e \cdot m \arrow[d, Rightarrow, "\mathsf{cell}_e"] \\
(\mathsf{mor}_e \cdot m) \cdot m \arrow[r, Rightarrow, "\mathsf{cell}_e \triangleright m"'] & (\mathrm{id}_x \cdot \mathsf{mor}_e) \cdot m \arrow[r, Rightarrow, "\lambda \triangleright m"'] & \mathsf{mor}_e \cdot m \arrow[r, Rightarrow, "\mathsf{cell}_e"'] & \mathrm{id}_x \cdot \mathsf{mor}_e
\end{tikzcd}$$

If no confusion arises, we write e instead of ob_e.

Next we look at the universal property of Eilenberg-Moore objects. Since Eilenberg-Moore objects are examples of limits in bicategories [43], there are multiple methods to express their universal property. A first possibility, which is used by Street, is to use biadjunctions [47], and a second option is to write out explicit mapping properties. Alternatively, one could express the universal property as an adjoint equivalence on the hom-categories. We use the last option. To write out the desired definition precisely, we first define a functor with domain $\underline{\mathsf{B}(x, e)}$ where e is an Eilenberg-Moore cone and x is any object.

▶ **Problem 5.2.** *Given an Eilenberg-Moore cone e for m and an object x, to construct a functor* $\mathsf{EMFunctor}_{x,e} : \underline{\mathsf{B}(x, e)} \to \underline{\mathsf{Mnd}(\mathsf{B})(\mathsf{idMnd}(x), m)}$.

▶ **Construction 5.3** (for Problem 5.2; `em_hom_functor`)**.** Suppose that we have a 1-cell $f : x \to e$. We construct the monad morphism $\mathsf{EMFunctor}_{x,e}(f)$ as follows:

- The underlying morphism is $f \cdot \mathsf{mor}_e$.
- For the 2-cell $(f \cdot \mathsf{mor}_e) \cdot m \Rightarrow \mathrm{id}_x \cdot (f \cdot \mathsf{mor}_e)$, we take

$$(f \cdot \mathsf{mor}_e) \cdot m \xrightarrow{\alpha^{-1}} f \cdot \mathsf{mor}_e \cdot m \xrightarrow{f \triangleleft \mathsf{cell}_e} f \cdot \mathrm{id}_e \cdot \mathsf{mor}_e \xrightarrow{f \triangleleft \lambda} f \cdot \mathsf{mor}_e \xrightarrow{\lambda^{-1}} \mathrm{id}_x \cdot (f \cdot \mathsf{mor}_e).$$

Given a 2-cell $\tau : f \Rightarrow g$, the underlying cell of $\mathsf{EMFunctor}_{x,e}(\tau) : \mathsf{EMFunctor}_{x,e}(f) \Rightarrow \mathsf{EMFunctor}_{x,e}(g)$ is $\tau \triangleright \mathsf{mor}_e$. ⌟

▶ **Definition 5.4** (`bicat_has_em`)**.** Let B be a bicategory and let m be a monad in B. An Eilenberg-Moore cone e for m is called **universal** if for every object x the functor $\mathsf{EMFunctor}_{x,e}$ is an adjoint equivalence of categories. We say that a bicategory has **Eilenberg-Moore objects** if for every monad m there is a universal Eilenberg-Moore cone.

▶ **Proposition 5.5** (`isaprop_is_universal_em_cone`)**.** *Let B be a locally univalent bicategory and let e be an Eilenberg-Moore cone for a monad m in B. The type that e is universal, is a proposition.*

Another way to formulate the universality of an Eilenberg-Moore cone, is by using Eilenberg-Moore categories. If we have a monad m on a category C, we write $\mathsf{EM}(m)$ for the Eilenberg-Moore category of m. Recall that the objects of $\mathsf{EM}(m)$ are pairs (x, f) of an object $x : \mathsf{C}$ and a morphism $f : m(x) \to x$ such that $\eta_m(x) \cdot f = \mathrm{id}_x$ and $\mu_m(x) \cdot f = m(f) \cdot f$. Morphisms from (x, f) to (y, g) are morphisms $h : x \to y$ such that $f \cdot h = m(h) \cdot g$.

▶ **Problem 5.6.** *Given an Eilenberg-Moore cone e for a monad m and an object $x : \mathsf{B}$, to construct a functor* $\mathsf{EMFunctor}'_{x,e} : \underline{\mathsf{B}(x, e)} \to \mathsf{EM}(\mathsf{HomMnd}_x(m))$.

▶ **Construction 5.7** (for Problem 5.6; `is_em_universal_em_cone_functor`)**.** First, we say how $\mathsf{EMFunctor}'_{x,e}$ acts on objects. Suppose that we have $f : x \to e$. To define an object of $\mathsf{EM}(\mathsf{HomMnd}_x(m))$, we first need to give a 1-cell $h : x \to \mathsf{ob}_x$. We define h as $x \xrightarrow{f} e \xrightarrow{\mathsf{mor}_e} \mathsf{ob}_m$ We also need to define a 2-cell $\tau : h \cdot m \Rightarrow h$ for which we take

$$(f \cdot \mathsf{mor}_e) \cdot m \xRightarrow{\alpha} f \cdot (\mathsf{mor}_e \cdot m) \xRightarrow{f \lhd \mathsf{cell}_e} f \cdot (\mathrm{id}_x \cdot \mathsf{mor}_e) \xRightarrow{f \lhd \lambda} f \cdot \mathsf{mor}_e$$

Next we define the action on morphisms. Suppose that we have $f, g : x \to e$ and a 2-cell $\theta : f \Rightarrow g$. We need to construct a 2-cell $f \cdot \mathsf{mor}_e \Rightarrow g \cdot \mathsf{mor}_e$, for which we take $\theta \rhd \mathsf{mor}_e$. ⌟

▶ **Proposition 5.8** (`is_universal_em_cone_weq_is_em_universal_em_cone`)**.** *Let e be an Eilenberg-Moore cone. Then e is universal if and only if for every x the functor* $\mathsf{EMFunctor}'_{x,e}$ *is an adjoint equivalence.*

Proof. We only give a brief sketch of the proof. The main idea is that one can construct an adjoint equivalence from $\mathsf{EM}(\mathsf{HomMnd}_x(m))$ to $\underline{\mathsf{Mnd}(\mathsf{B})(\mathsf{idMnd}(x), m)}$. With this equivalence available, the theorem follows from the 2-out-of-3 property. ◀

Alternatively, universality can be formulated using mapping properties. These properties are deduced from the fact that every adjoint equivalence is split essentially surjective and fully faithful. More precisely, an Eilenberg-Moore cone e is universal if and only if

- for every Eilenberg-Moore cone q there is a 1-cell $\mathsf{EM}_{\mathsf{mor}}(q) : q \to e$ and an invertible 2-cell $\mathsf{EM}_{\mathsf{com}}(q) : \mathsf{idMnd}(\mathsf{EM}_{\mathsf{mor}}(q)) \cdot \mathsf{mor}_e \Rightarrow \mathsf{mor}_q$ in $\mathsf{Mnd}(\mathsf{B})$;
- for all 1-cells $g_1, g_2 : q \to e$ and 2-cells $\tau : \mathsf{idMnd}(g_1) \cdot \mathsf{mor}_e \Rightarrow \mathsf{idMnd}(g_2) \cdot \mathsf{mor}_e$ in $\mathsf{Mnd}(\mathsf{B})$, there is a unique 2-cell $\mathsf{EM}_{\mathsf{cell}}(\tau)$ such that $\mathsf{idMnd}(\mathsf{EM}_{\mathsf{cell}}(\tau)) \rhd \mathsf{mor}_e = \tau$.

Let us consider some examples of Eilenberg-Moore objects.

▶ **Example 5.9** (`has_em_bicat_of_univ_cats`)**.** The bicategory $\mathsf{UnivCat}$ has Eilenberg-Moore objects. As suggested by the name, the Eilenberg-Moore object on a monad m is indeed given by the Eilenberg-Moore category $\mathsf{EM}(m)$. Note that if C is a univalent and m is a monad on C, then $\mathsf{EM}(m)$ is univalent as well.

▶ **Example 5.10** (`has_em_univ_cat_with_terminal_obj`)**.** The bicategory $\mathsf{UnivCat}_{\mathsf{Terminal}}$ has Eilenberg-Moore objects as well. If we have a monad $m : \mathsf{Mnd}(\mathsf{UnivCat}_{\mathsf{Terminal}})$, then ob_m has a terminal object and m preserves terminal objects. Under these conditions, it follows that the Eilenberg-Moore category of m has a terminal object. From this, we can conclude that $\mathsf{UnivCat}_{\mathsf{Terminal}}$ indeed has Eilenberg-Moore objects.

One can also show that $\mathsf{SymMonUnivCat}^{\mathsf{co}}$ has Eilenberg-Moore objects. These are given by Eilenberg-Moore categories of comonads.

6 Duality and Kleisli Objects

The goal of this section is to construct Eilenberg-Moore objects in $\mathsf{UnivCat}^{\mathsf{op}}$. To do so, we start by characterizing such objects via *Kleisli objects*.

▶ **Definition 6.1** (`kleisli_cocone`)**.** Let B be a bicategory and let m be a monad in B. A **Kleisli cocone** k for m in B consists of an object $\mathsf{ob}_k : \mathsf{B}$, a 1-cell $\mathsf{mor}_k : \mathsf{ob}_m \to \mathsf{ob}_k$, and a 2-cell $\mathsf{cell}_k : m \cdot \mathsf{mor}_k \Rightarrow \mathsf{mor}_k$ such that the following diagrams commute

$$\begin{tikzcd} \mathrm{id}_{\mathsf{ob}_m} \cdot \mathsf{mor}_k \arrow[r, Rightarrow, "\eta_m \rhd \mathsf{mor}_k"] \arrow[dr, Rightarrow, "\lambda"'] & m \cdot \mathsf{mor}_k \arrow[d, Rightarrow, "\mathsf{cell}_k"] \\ & \mathsf{mor}_k \end{tikzcd}$$

$$\begin{tikzcd} (m \cdot m) \cdot \mathsf{mor}_k \arrow[r, Rightarrow, "\alpha^{-1}"] \arrow[d, Rightarrow, "\mu_m \rhd \mathsf{mor}_k"'] & m \cdot (m \cdot \mathsf{mor}_k) \arrow[r, Rightarrow, "m \lhd \mathsf{cell}_k"] & m \cdot \mathsf{mor}_k \arrow[d, Rightarrow, "\mathsf{cell}_k"] \\ m \cdot \mathsf{mor}_k \arrow[rr, Rightarrow, "\mathsf{cell}_k"'] & & \mathsf{mor}_k \end{tikzcd}$$

▶ **Definition 6.2** (`has_kleisli_ump`)**.** A Kleisli cocone k is said to be **universal** if

- For every Kleisli cocone q there there is a 1-cell $\mathsf{Kl}_{\mathsf{mor}}(q) : \mathsf{ob}_k \to \mathsf{ob}_q$ and an invertible 2-cell $\mathsf{Kl}_{\mathsf{com}}(q) : \mathsf{mor}_k \cdot \mathsf{Kl}_{\mathsf{mor}}(q) \Rightarrow \mathsf{mor}_q$ such that the following diagram commutes

$$\begin{tikzcd} m \cdot (\mathsf{mor}_k \cdot \mathsf{Kl}_{\mathsf{mor}}(q)) \arrow[rr, Rightarrow, "m \lhd \mathsf{Kl}_{\mathsf{com}}(q)"] \arrow[d, Rightarrow, "\alpha"'] & & m \cdot \mathsf{mor}_q \arrow[d, Rightarrow, "\mathsf{cell}_q"] \\ (m \cdot \mathsf{mor}_k) \cdot \mathsf{Kl}_{\mathsf{mor}}(q) \arrow[r, Rightarrow, "\mathsf{cell}_k \rhd \mathsf{Kl}_{\mathsf{mor}}(q)"'] & \mathsf{mor}_k \cdot \mathsf{Kl}_{\mathsf{mor}}(q) \arrow[r, Rightarrow, "\mathsf{Kl}_{\mathsf{com}}(q)"'] & \mathsf{mor}_q \end{tikzcd}$$

- Suppose that we have an object $x : \mathsf{B}$, two 1-cells $g_1, g_2 : \mathsf{ob}_k \to x$, and a 2-cell $\tau : \mathsf{mor}_k \cdot g_1 \Rightarrow \mathsf{mor}_k \cdot g_2$ such that the following diagram commutes

$$\begin{tikzcd} m \cdot (\mathsf{mor}_k \cdot g_1) \arrow[r, Rightarrow, "\alpha"] \arrow[d, Rightarrow, "m \lhd \tau"'] & (m \cdot \mathsf{mor}_k) \cdot g_1 \arrow[r, Rightarrow, "\mathsf{cell}_k \rhd g_1"] & \mathsf{mor}_k \cdot g_1 \arrow[d, Rightarrow, "\tau"] \\ m \cdot (\mathsf{mor}_k \cdot g_2) \arrow[r, Rightarrow, "\alpha"'] & (m \cdot \mathsf{mor}_k) \cdot g_2 \arrow[r, Rightarrow, "\mathsf{cell}_k \rhd g_2"'] & \mathsf{mor}_k \cdot g_2 \end{tikzcd}$$

  Then there is a unique 2-cell $\mathsf{Kl}_{\mathsf{cell}}(\tau) : g_1 \Rightarrow g_2$ such that $\mathsf{cell}_k \lhd \mathsf{Kl}_{\mathsf{cell}}(\tau) = \tau$.

A **Kleisli object** is a universal Kleisli cocone. We say that a bicategory has Kleisli objects if there is a Kleisli object for every monad m.

▶ **Proposition 6.3** (`op1_has_em`)**.** *If* B *has Kleisli objects, then* B^{op} *has Eilenberg-Moore objects.*

As such, to find Eilenberg-Moore objects in $\mathsf{UnivCat}^{\mathsf{op}}$, we need to find Kleisli objects in $\mathsf{UnivCat}$. However, before we look at those, we look at Kleisli objects in Cat. These are constructed via the usual definition of *Kleisli categories*.

▶ **Problem 6.4.** *To construct Kleisli objects in* Cat*.*

▶ **Construction 6.5** (for Problem 6.4; `bicat_of_cats_has_kleisli`)**.** Recall that given a monad m on a category C, the Kleisli category $\mathcal{K}(m)$ is usually defined to be the category whose objects are $x : \mathsf{C}$ and whose morphisms from $x : \mathsf{C}$ to $y : \mathsf{C}$ are morphisms $x \to m(y)$.

Note that we have a functor $F : \mathsf{C} \to \mathcal{K}(m)$: it sends objects x to x and morphisms $f : x \to y$ to $x \xrightarrow{f} y \xrightarrow{\eta_m(y)} m(y)$. We also have a natural transformation $m \cdot F \Rightarrow F$, which is the identity on ever object x. As such, we have a Kleisli cocone. This cocone is universal, and for a proof we refer the reader to the formalization. ⌟

Note that even if C is required to be univalent, the Kleisli category $\mathcal{K}(m)$ is *not* necessarily univalent. As such, to obtain Kleisli objects in $\mathsf{UnivCat}$, we need to use an alternative definition for the Kleisli category [6]. First, we define a functor $\mathsf{FreeAlg}_m : \mathsf{C} \to \mathsf{EM}(m)$ which sends objects $x : \mathsf{C}$ to the algebra $\mu_m(x) : m(m(x)) \to m(x)$ and morphisms $f : x \to y$ to $m(f) : m(x) \to m(y)$. Note that this 1-cell can actually be defined in arbitrary bicategories (see Section 7). By taking the full image of this functor, we obtain the category $\mathsf{Kleisli}(m)$.

- Objects of $\mathsf{Kleisli}(m)$ are pairs $y : \mathsf{C}$ together with a proof of $\|\sum(x : \mathsf{C}), m(x) \cong y\|$.
- Morphisms from $y_1 : \mathsf{Kleisli}(m)$ to $y_2 : \mathsf{Kleisli}(m)$ are morphisms $y_1 \to y_2$ in C.

If C is univalent, then $\mathsf{EM}_{\mathsf{UnivCat}}(m)$ is univalent, and thus $\mathsf{Kleisli}(m)$ is so as well.

▶ **Problem 6.6.** *To construct a fully faithful essentially surjective functor* $\mathsf{incl}_m : \mathcal{K}(m) \to \mathsf{Kleisli}(m)$.

▶ **Construction 6.7** (for Problem 6.6; `functor_to_kleisli_cat`)**.** The functor incl_m sends every object $x : \mathcal{K}(m)$ to $\mathsf{FreeAlg}_m(x)$, which is indeed in the image of $\mathsf{FreeAlg}_m$. Morphisms $f : x \to m(y)$ are sent to $m(x) \xrightarrow{m(f)} m(m(y)) \xrightarrow{\mu_m(y)} m(y)$. A proof that this functor is both essentially surjective and fully faithful can be found in the formalization. ⌟

In univalent foundations, not every functor that is both fully faithful and essentially surjective is automatically an adjoint equivalence as well. This statement only holds if the domain is univalent. However, we can still use the functor incl_m to deduce the universal property of $\mathsf{Kleisli}(m)$. For that, we use Theorem 8.4 in [2].

▶ **Proposition 6.8** (`precomp_adjoint_equivalence`)**.** *Let $F : \mathsf{C}_1 \to \mathsf{C}_2$ be a fully faithful and essentially surjective functor, and suppose that C_3 is a univalent category. Then the functor $(F \cdot -)_{\mathsf{C}_3} : \mathsf{C}_3^{\mathsf{C}_2} \to \mathsf{C}_3^{\mathsf{C}_1}$, given by precomposition with F, is an adjoint equivalence.*

▶ **Problem 6.9.** *To construct Kleisli objects in* $\mathsf{UnivCat}$.

▶ **Construction 6.10** (for Problem 6.9; `bicat_of_univ_cats_has_kleisli`)**.** We only show how to construct the required 1-cells. Suppose that we have a Kleisli cocone q in $\mathsf{UnivCat}$. Note that q also is a Kleisli cocone in Cat, and as such, we get a functor $\mathsf{Kl}_{\mathsf{mor}}(q) : \mathcal{K}(m) \to \mathsf{ob}_q$. By Proposition 6.8, we now get the desired functor $\mathsf{Kleisli}(m) \to \mathsf{ob}_q$. ⌟

7 Monads and Adjunctions

The cornerstone of the theory of monads is the relation between monads and adjunctions. More specifically, every adjunction gives rise to a monad and vice versa. This was generalized by Street to 2-categories that have Eilenberg-Moore objects [47]. In this section, we prove these theorems, and to do so, we start by recalling adjunctions in bicategories.

▶ **Definition 7.1** (`adjunction`)**.** An **adjunction** $l \frac{\varepsilon}{\eta}\Big| r$ in a bicategory B consists of

- objects x and y;
- 1-cells $l : x \to y$ and $r : y \to x$;
- 2-cells $\eta : \mathrm{id}_x \Rightarrow l \cdot r$ and $\varepsilon : r \cdot l \Rightarrow \mathrm{id}_y$

such that the following 2-cells are identities

$$l \xrightarrow{\lambda^{-1}} \mathrm{id}_x \cdot l \xrightarrow{\eta \triangleright l} (l \cdot r) \cdot l \xrightarrow{\alpha^{-1}} l \cdot (r \cdot l) \xrightarrow{l \triangleleft \varepsilon} l \cdot \mathrm{id}_y \xrightarrow{\rho} l$$

$$r \xrightarrow{\rho^{-1}} r \cdot \mathrm{id}_x \xrightarrow{r \triangleleft \eta} r \cdot (l \cdot r) \xrightarrow{\alpha} (r \cdot l) \cdot r \xrightarrow{\varepsilon \triangleright r} \mathrm{id}_y \cdot r \xrightarrow{\lambda} r$$

Our notation for adjunctions is taken from [15]. The two coherences in Definition 7.1 are called the *triangle equalities*. As expected, adjunctions internal to $\mathsf{UnivCat}$ correspond to adjunctions of categories [31]. This is because the unitors and associators in $\mathsf{UnivCat}$ are pointwise the identity, so the triangle equalities in Definition 7.1 reduce to the usual ones.

▶ **Example 7.2.** We characterize adjunctions in B^{op} and B^{co} as follows.

- (`op1_left_adjoint_to_right_adjoint`) Every adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$ in B gives rise to an adjunction $r \frac{\varepsilon}{\eta}\!\!\mid l$ in B^{op} and vice versa.
- (`op2_left_adjoint_to_right_adjoint`) Every adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$ in B gives rise to an adjunction $r \frac{\eta}{\varepsilon}\!\!\mid l$ in B^{co} and vice versa.

Example 7.2 can be strengthened by using the terminology of *left adjoints* and *right adjoints*. Given a 1-cell $f : x \to y$, the type $\mathsf{LeftAdj}_{\mathsf{B}}(f)$ says that we have r,η, and ε such that we have an adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$. The type $\mathsf{RightAdj}_{\mathsf{B}}(f)$ is defined analogously. Now we can reformulate Example 7.2 as follows: we have equivalences $\mathsf{LeftAdj}_{\mathsf{B}^{\mathsf{op}}}(f) \simeq \mathsf{RightAdj}_{\mathsf{B}}(f)$ and $\mathsf{LeftAdj}_{\mathsf{B}^{\mathsf{co}}}(f) \simeq \mathsf{RightAdj}_{\mathsf{B}}(f)$ of types.

Next we look at *displayed adjunctions*, which we use to obtain adjunctions in total bicategories [1]. This notion is used to characterize adjunctions in bicategories such as $\mathsf{UnivCat}_{\mathsf{Terminal}}$ and $\mathsf{SymMonUnivCat}$.

▶ **Definition 7.3** (`disp_adjunction`)**.** Let B be a bicategory and let D be a displayed bicategory over B. A **displayed adjunction** over an adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$ where $l : x \to y$ consists of

- Objects $\overline{x} : D_x$ and $\overline{y} : D_y$;
- Displayed morphisms $\overline{l} : \overline{x} \xrightarrow{l} \overline{y}$ and $\overline{r} : \overline{y} \xrightarrow{r} \overline{x}$;
- Displayed 2-cells $\overline{\eta} : \overline{\mathrm{id}_x} \xRightarrow{\eta} \overline{l} \cdot \overline{r}$ and $\overline{\varepsilon} : \overline{r} \cdot \overline{l} \xRightarrow{\varepsilon} \mathrm{id}_x$.

We also require some coherences and those can be found in the formalization. We denote this data by $\overline{l} \frac{\overline{\varepsilon}}{\overline{\eta}}\!\!\mid \overline{r}$.

▶ **Problem 7.4.** *Given a displayed adjunction* $\overline{l} \frac{\overline{\varepsilon}}{\overline{\eta}}\!\!\mid \overline{r}$ *in a displayed bicategory* D *over* $l \frac{\varepsilon}{\eta}\!\!\mid r$*, to construct an adjunction* $\int \overline{l} \frac{\overline{\varepsilon}}{\overline{\eta}}\!\!\mid \overline{r}$ *in* $\int D$*.*

▶ **Construction 7.5** (for Problem 7.4; `left_adjoint_data_total_weq`)**.** The left adjoint of $\int \overline{l} \frac{\overline{\varepsilon}}{\overline{\eta}}\!\!\mid \overline{r}$ is $(l, \overline{l})$, the right adjoint is $(r, \overline{r})$, the unit is $(\eta, \overline{\eta})$, and the counit is $(\varepsilon, \overline{\varepsilon})$. ⌟

▶ **Example 7.6** (`disp_adj_weq_preserves_terminal`)**.** Adjunctions in $\mathsf{UnivCat}_{\mathsf{Terminal}}$ are given by an adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$ in $\mathsf{UnivCat}$ such that l preserves terminal objects. Note that r automatically preserves terminal objects, because r is a right adjoint.

Analogously, we characterize adjunctions in $\mathsf{SymMonUnivCat}$. Now we have developed enough to state and prove the core theorems of the formal theory of monads [47]. These theorems relate adjunctions and monads, and we first prove that every adjunction gives rise to a monad.

▶ **Problem 7.7.** *Given an adjunction* $l \frac{\varepsilon}{\eta}\!\!\mid r$*, to construct a monad* $\mathsf{AdjToMnd}(l \frac{\varepsilon}{\eta}\!\!\mid r)$*.*

▶ **Construction 7.8** (for Problem 7.7; `mnd_from_adjunction`). Let an adjunction $l \frac{\varepsilon}{\eta}\!\!\mid r$ be given where $l : x \to y$. We define the monad $\mathsf{AdjToMnd}(l \frac{\varepsilon}{\eta}\!\!\mid r)$ as follows.

- Its object is x;
- The endomorphism is $l \cdot r : x \to x$;
- The unit is $\eta : \mathrm{id}_x \to l \cdot r$;
- For the multiplication, we use the following composition of 2-cells

$$(l \cdot r) \cdot (l \cdot r) \xRightarrow{\alpha} l \cdot (r \cdot (l \cdot r)) \xRightarrow{l \triangleleft \alpha^{-1}} l \cdot ((r \cdot l) \cdot r) \xRightarrow{l \triangleleft (\varepsilon \triangleright r)} l \cdot (\mathrm{id}_y \cdot r) \xRightarrow{l \triangleleft \lambda} l \cdot r$$

The proofs of the necessary equalities can be found in the formalization. ⌟

Since by Example 4.5 and 7.2 adjunctions and monads in B^{co} correspond to adjunctions and comonads in B respectively, we get that every adjunction in B induces a comonad by Construction 7.8. Next we look at the converse: obtaining adjunctions from monads. For this, we need to work in a bicategory with Eilenberg-Moore objects. We show that every monad m gives rise to an adjunction and that the monad coming from this adjunction is equivalent to m.

▶ **Problem 7.9.** *Given a bicategory* B *with Eilenberg-Moore objects and a monad* m *in* B*, to construct an adjunction* $\mathsf{MndToAdj}(m)$ *and an adjoint equivalence* $\mathsf{MndEquiv}(m)$ *between* $\mathsf{AdjToMnd}(\mathsf{MndToAdj}(m))$ *and* m*.*

▶ **Construction 7.10** (for Problem 7.9; `mnd_to_adjunction`). The right adjoint is the 1-cell $\mathsf{mor}_e : \mathsf{ob}_m \to \mathsf{EM}_{\mathsf{B}}(m)$. For the left adjoint, we need to define a 1-cell $\mathsf{FreeAlg}_m : \mathsf{EM}_{\mathsf{B}}(m) \to \mathsf{ob}_m$, and we use the universal property of Eilenberg-Moore objects for that. We construct a cone q as follows:

- The object is ob_m;
- The morphism is m;
- The 2-cell is $m \cdot m \xRightarrow{\mu_m} m \xRightarrow{\lambda^{-1}} \mathrm{id}_{\mathsf{ob}_m} \cdot m$.

We define $\mathsf{FreeAlg}_m$ as $\mathsf{EM}_{\mathsf{mor}}(q)$. The unit of the desired adjunction is defined as follows:

$$\mathrm{id}_{\mathsf{ob}_m} \xRightarrow{\eta_m} m \xRightarrow{\mathsf{EM}_{\mathsf{com}}(q)} \mathsf{FreeAlg}_m \cdot \mathsf{mor}_e$$

For the counit we use the universal property of Eilenberg-Moore objects. The details can be found in the formalization.

To construct $\mathsf{MndEquiv}(m)$, we use Proposition 3.7. Hence, it suffices to construct a monad morphism $G : \mathsf{AdjToMnd}(\mathsf{MndToAdj}(m)) \to m$ whose underlying 1-cell and 2-cell are an adjoint equivalence and invertible respectively. We define G as follows:

- The underlying 1-cell is $\mathrm{id}_{\mathsf{ob}_m} : \mathsf{ob}_m \to \mathsf{ob}_m$.
- For the underlying 2-cell, we take

$$\mathrm{id}_{\mathsf{ob}_m} \cdot m \xRightarrow{\lambda} m \xRightarrow{\mathsf{EM}_{\mathsf{com}}(q)} \mathsf{FreeAlg}_m \cdot \mathsf{mor}_e \xRightarrow{\rho^{-1}} (\mathsf{FreeAlg}_m \cdot \mathsf{mor}_e) \cdot \mathrm{id}_{\mathsf{ob}_m}$$ ⌟

Note that we can instantiate Construction 7.10 to several concrete instances.

- Since $\mathsf{UnivCat}$ has Eilenberg-Moore objects by Example 5.9, we get the usual construction of adjunctions from monads via Eilenberg-Moore categories.
- Since $\mathsf{UnivCat}^{\mathsf{op}}$ has Eilenberg-Moore objects by Construction 6.10, every monad gives rise to an adjunction via Kleisli categories.

One can also show that $\mathsf{SymMonUnivCat}^{\mathsf{co}}$ has Eilenberg-Moore objects, and thus every comonad of symmetric monoidal categories gives rise to an adjunction.

8 Monadic Adjunctions

By Construction 7.10 we have an equivalence $\mathsf{AdjToMnd}(\mathsf{MndToAdj}(m)) \simeq m$ for every monad m. However, such a statement does not hold for adjunctions. *Monadic adjunctions* are the adjunctions for which we do have such an equivalence.

▶ **Problem 8.1.** *Given an adjunction $l \underset{\eta}{\overset{\varepsilon}{\dashv}} r$ where $l : x \to y$ in a bicategory B with Eilenberg-Moore objects, to construct a 1-cell $\mathsf{Comparison}(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r) : y \to \mathsf{EM}_{\mathsf{B}}(\mathsf{AdjToMnd}(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r))$.*

▶ **Construction 8.2** (for Problem 8.1; `comparison_mor`)**.** We use the universal property of Eilenberg-Moore objects, and we construct a cone q as follows
- The object is y;
- The 1-cell is $r : y \to x$;
- The 2-cell is $r \cdot (l \cdot r) \xRightarrow{\alpha} (r \cdot l) \cdot r \xRightarrow{\varepsilon \triangleright r} \mathrm{id}_y \cdot r$.

Note that the object and 1-cell of $\mathsf{AdjToMnd}(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r)$ are x and $l \cdot r$ respectively. Now we define $\mathsf{Comparison}(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r)$ to be $\mathsf{EM}_{\mathsf{mor}}(q)$. ⌟

▶ **Definition 8.3** (`is_monadic`)**.** An adjunction $l \underset{\eta}{\overset{\varepsilon}{\dashv}} r$ in a bicategory B with Eilenberg-Moore objects is called **monadic** if the 1-cell $\mathsf{Comparison}(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r)$ is an adjoint equivalence.

Next we look at a representable version of this definition. More specifically, we define monadic 1-cells using monadic functors in $\mathsf{UnivCat}$. To do so, we first show that every adjunction gives rise to an adjunction on the hom-categories.

▶ **Problem 8.4.** *Given $l \underset{\eta}{\overset{\varepsilon}{\dashv}} r$ where $l : x \to y$ and an object w, to construct an adjunction $\mathsf{HomAdj}_w(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r)$ between $\underline{\mathsf{B}(w,x)}$ and $\underline{\mathsf{B}(w,y)}$.*

▶ **Construction 8.5** (for Problem 8.4; `left_adjoint_to_adjunction_cat`)**.** The left adjoint is $(- \cdot l)_w$, while the right adjoint is $(- \cdot r)_w$. For the unit, we need to construct natural 2-cells $f \Rightarrow (f \cdot l) \cdot r$, and for which we take

$$f \xRightarrow{\rho^{-1}} f \cdot \mathrm{id}_y \xRightarrow{f \triangleleft \eta} f \cdot (l \cdot r) \xRightarrow{\alpha} (f \cdot l) \cdot r$$

For the counit, we construct natural 2-cells $(f \cdot r) \cdot l$, which are defined as follows

$$(f \cdot r) \cdot l \xRightarrow{\alpha^{-1}} f \cdot (r \cdot l) \xRightarrow{f \triangleleft \varepsilon} f \cdot \mathrm{id}_y \xRightarrow{\rho} f$$

⌟

▶ **Definition 8.6** (`is_monadic_repr`)**.** An adjunction $l \underset{\eta}{\overset{\varepsilon}{\dashv}} r$ in a locally univalent bicategory is called **representably monadic** if for every $w \in \mathsf{B}$ the adjunction $\mathsf{HomAdj}_w(l \underset{\eta}{\overset{\varepsilon}{\dashv}} r)$ is a monadic 1-cell in $\mathsf{UnivCat}$.

Note that we require the bicategory in Definition 8.6 to be locally univalent, so that each hom-category lies in $\mathsf{UnivCat}$. Now we show that these two notions of monadicity are equivalent. We first prove the following lemma.

▶ **Lemma 8.7** (`left_adjoint_equivalence_weq_left_adjoint_equivalence_repr`)**.** *A 1-cell $f : x \to y$ is an adjoint equivalence if and only if for all w the functor $(- \cdot f)_w$ is an adjoint equivalence of categories.*

Proof. Suppose, we have a a adjoint equivalence $f : x \simeq y$ and let $w : \mathsf{B}$. By Construction 8.5, we obtain an adjunction $\mathsf{HomAdj}_w(l \overset{\varepsilon}{\underset{\eta}{\dashv}} r)$ between $\underline{\mathsf{B}(w,x)}$ and $\underline{\mathsf{B}(w,y)}$ whose right adjoint is $(- \cdot f)_w$. Since the unit and counit of f are invertible, the unit and counit of $\mathsf{HomAdj}_w(l \overset{\varepsilon}{\underset{\eta}{\dashv}} r)$ are invertible as well, and thus we get the desired adjoint equivalence.

Next suppose that for every w the functor $(- \cdot f)_w$ is an adjoint equivalence. For every w, we denote its right adjoint by $R_w : \underline{\mathsf{B}(w,y)} \to \underline{\mathsf{B}(w,x)}$, its unit by $\eta_w : R_w \cdot (- \cdot f)_w \Rightarrow \mathrm{id}$, and its counit by $\varepsilon_w : \mathrm{id} \Rightarrow (- \cdot f)_w \cdot R_w$. Now we show that f is an adjoint equivalence

- The right adjoint is $R_y(\mathrm{id}_y) : \underline{\mathsf{B}(y,x)}$.
- The unit is $\eta_w(\mathrm{id}_y) : R_y(\mathrm{id}_y) \cdot f \Rightarrow \mathrm{id}_y$.
- For the counit, we use the following composition

$$
\begin{tikzcd}
f \cdot R_y(\mathrm{id}_y) \arrow[r, Rightarrow, "\eta_x(f \cdot R_y(\mathrm{id}_y))"] & R_x((f \cdot R_y(\mathrm{id}_y)) \cdot f) \arrow[r, Rightarrow, "R(\alpha^{-1})"] & R_x(f \cdot (R_y(\mathrm{id}_y) \cdot f)) \arrow[dll, Rightarrow, "R(f \lhd \eta_y(\mathrm{id}_y))"'] & \\
R_x(f \cdot \mathrm{id}_y) \arrow[r, Rightarrow, "R(\rho)"'] & R_x(f) \arrow[r, Rightarrow, "R(\lambda^{-1})"'] & R_x(\mathrm{id}_x \cdot f) \arrow[r, Rightarrow, "\eta_x^{-1}(\mathrm{id}_x)"'] & \mathrm{id}_x
\end{tikzcd}
$$

Since both the unit and counit are invertible, f is indeed an adjoint equivalence. ◀

▶ **Theorem 8.8** (`is_monadic_repr_weq_is_monadic`)**.** *An adjunction is monadic if and only if it is representably monadic.*

Proof. Suppose that we have an adjunction $l \overset{\varepsilon}{\underset{\eta}{\dashv}} r$ and that we have $w : \mathsf{B}$. First, we note that we have a monad on $\underline{\mathsf{B}(w,y)}$, namely $m := \mathsf{HomMnd}_w(\mathsf{HomAdj}_w(l \overset{\varepsilon}{\underset{\eta}{\dashv}} r))$. We denote the comparison cell $\mathsf{Comparison}(\mathsf{HomAdj}_w(l \overset{\varepsilon}{\underset{\eta}{\dashv}} r))$ by F.

We also have a functor $(- \cdot \mathsf{Comparison}(l \overset{\varepsilon}{\underset{\eta}{\dashv}} r))_w : \underline{\mathsf{B}(w,y)} \to \underline{\mathsf{B}(w, \mathsf{EM}_{\mathsf{B}}(m'))}$, which we denote by G. We write m' for the monad $\mathsf{HomMnd}_w(m)$, and recall that by Proposition 5.8 we have an adjoint equivalence from $\underline{\mathsf{B}(w, \mathsf{EM}_{\mathsf{B}}(m'))}$ to $\mathsf{EM}(\mathsf{HomMnd}_w(m))$. Denote this equivalence by H. There also is a functor $K : \mathsf{EM}(m') \to \mathsf{EM}(m)$ and a natural isomorphism $\tau : F \cdot (H \cdot K) \cong G$: their precise definition can be found in the formalization.

As such, we have the following diagram for every $w : \mathsf{B}$:

$$
\begin{tikzcd}
\underline{\mathsf{B}(w,y)} \arrow[r, "G"] \arrow[d, "F"'] & \mathsf{EM}(m) \\
\underline{\mathsf{B}(w, \mathsf{EM}_{\mathsf{B}}(m'))} \arrow[r, "H"'] \arrow[ur, phantom, "\Uparrow \tau"] & \mathsf{EM}(m') \arrow[u, "K"']
\end{tikzcd}
$$

Since both H and K are adjoint equivalences, we deduce that F is an adjoint equivalence if and only if G is. As such, if $l \overset{\varepsilon}{\underset{\eta}{\dashv}} r$ is representably monadic, then F is an adjoint equivalence and thus G is an adjoint equivalence. From Lemma 8.7 we get that $l \overset{\varepsilon}{\underset{\eta}{\dashv}} r$ is monadic. For the converse, we use the same argument: if $l \overset{\varepsilon}{\underset{\eta}{\dashv}} r$ is monadic, then G is an adjoint equivalence. Hence, F is an adjoint equivalence as well, so $l \overset{\varepsilon}{\underset{\eta}{\dashv}} r$ is representably monadic. ◀

9 Conclusion

We developed Street's formal theory of monads in this paper. We saw that it provides a good setting to study monads in univalent foundations, because it allows us to prove the core theorems in arbitrary bicategories instead of only for categories. For that reason, it helps us with concrete problems, such as constructing an adjunction from a monad using the univalent version of the Kleisli category. This is because one only needs to prove a universal property instead of reproducing the whole construction of the adjunction.

There are numerous ways to continue this line of research. One result that is missing, is Theorem 12 from Street's paper [47]. In addition, the work in this setting provides a framework in which one can study numerous applications, such as models of linear logic [11, 34], Moggi-style semantics [36], call-by-push-value [30], and the enriched effect calculus [16]. Formalizing these applications would be a worthwhile extension. Finally, one could study extensions of the formal theory to a wider class of monads, such as graded monads [17] or relative monads [8].

References

1. Benedikt Ahrens, Dan Frumin, Marco Maggesi, Niccolò Veltri, and Niels van der Weide. Bicategories in univalent foundations. *Mathematical Structures in Computer Science*, pages 1–38, 2022. `doi:10.1017/S0960129522000032`.
2. Benedikt Ahrens, Krzysztof Kapulkin, and Michael Shulman. Univalent categories and the Rezk completion. *Mathematical Structures in Computer Science*, 25:1010–1039, 2015. `doi:10.1017/S0960129514000486`.
3. Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed Categories. *Logical Methods in Computer Science*, 15(1), 2019. `doi:10.23638/LMCS-15(1:20)2019`.
4. Benedikt Ahrens, Ralph Matthes, and Anders Mörtberg. From Signatures to Monads in UniMath. *J. Autom. Reason.*, 63(2):285–318, 2019. `doi:10.1007/s10817-018-9474-4`.
5. Benedikt Ahrens, Ralph Matthes, and Anders Mörtberg. Implementing a category-theoretic framework for typed abstract syntax. In Andrei Popescu and Steve Zdancewic, editors, *CPP '22: 11th ACM SIGPLAN International Conference on Certified Programs and Proofs, Philadelphia, PA, USA, January 17 - 18, 2022*, pages 307–323. ACM, 2022. `doi:10.1145/3497775.3503678`.
6. Benedikt Ahrens, Paige Randall North, Michael Shulman, and Dimitris Tsementzis. The univalence principle. *CoRR*, abs/2102.06275, 2021. `arXiv:2102.06275`.
7. Benedikt Ahrens, Paige Randall North, and Niels van der Weide. Semantics for two-dimensional type theory. *CoRR*, abs/2201.10662, 2022. `arXiv:2201.10662`.
8. Nathanael Arkor and Dylan McDermott. The formal theory of relative monads. *arXiv preprint arXiv:2302.14014*, 2023.
9. Andrej Bauer, Jason Gross, Peter LeFanu Lumsdaine, Michael Shulman, Matthieu Sozeau, and Bas Spitters. The HoTT library: a formalization of homotopy type theory in Coq. In Yves Bertot and Viktor Vafeiadis, editors, *Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, Paris, France, January 16-17, 2017*, pages 164–172. ACM, 2017. `doi:10.1145/3018610.3018615`.
10. Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar*, pages 1–77, Berlin, Heidelberg, 1967. Springer Berlin Heidelberg. `doi:10.1007/BFb0074299`.
11. P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models (extended abstract). In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, volume 933 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 1994. `doi:10.1007/BFb0022251`.
12. Gabriella Böhm, Stephen Lack, and Ross Street. On the 2-categories of weak distributive laws. *Communications in Algebra*, 39(12):4567–4583, 2011.
13. Matteo Capucci and Bruno Gavranović. Actegories for the Working Amthematician. *arXiv preprint arXiv:2203.16351*, 2022.
14. Eugenia Cheng. Iterated distributive laws. *Mathematical Proceedings of the Cambridge Philosophical Society*, 150(3):459–487, 2011.

15 Ivan Di Liberti and Fosco Loregian. On the unicity of formal category theories. *arXiv preprint arXiv:1901.01594*, 2019.

16 Jeff Egger, Rasmus Ejlers Møgelberg, and Alex Simpson. The enriched effect calculus: syntax and semantics. *J. Log. Comput.*, 24(3):615–654, 2014. doi:10.1093/logcom/exs025.

17 Soichiro Fujii, Shin-ya Katsumata, and Paul-André Melliès. Towards a Formal Theory of Graded Monads. In Bart Jacobs and Christof Löding, editors, *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9634 of *Lecture Notes in Computer Science*, pages 513–530. Springer, 2016. doi:10.1007/978-3-662-49630-5_30.

18 Jason Gross, Adam Chlipala, and David I. Spivak. Experience Implementing a Performant Category-Theory Library in Coq. In Gerwin Klein and Ruben Gamboa, editors, *Interactive Theorem Proving - 5th International Conference, ITP 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8558 of *Lecture Notes in Computer Science*, pages 275–291. Springer, 2014. doi:10.1007/978-3-319-08970-6_18.

19 Jason Z. S. Hu and Jacques Carette. Formalizing category theory in Agda. In Catalin Hritcu and Andrei Popescu, editors, *CPP '21: 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, Virtual Event, Denmark, January 17-19, 2021*, pages 327–342. ACM, 2021. doi:10.1145/3437992.3439922.

20 Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electronic Notes in Theoretical Computer Science*, 172:437–458, 2007.

21 Niles Johnson and Donald Yau. *2-dimensional categories.* Oxford University Press, USA, 2021.

22 Simon L. Peyton Jones and Philip Wadler. Imperative Functional Programming. In Mary S. Van Deusen and Bernard Lang, editors, *Conference Record of the Twentieth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Charleston, South Carolina, USA, January 1993*, pages 71–84. ACM Press, 1993. doi:10.1145/158511.158524.

23 Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The Simplicial Model of Univalent Foundations (after Voevodsky). *Journal of the European Mathematical Society*, 23(6):2071–2126, 2021. doi:10.4171/JEMS/1050.

24 Gregory Maxwell Kelly. Elementary observations on 2-categorical limits. *Bulletin of the Australian Mathematical Society*, 39(2):301–317, 1989.

25 Max Kelly. *Basic concepts of enriched category theory*, volume 64. CUP Archive, 1982.

26 Stephen Lack. Composing props. *Theory and Applications of Categories*, 13(9):147–163, 2004.

27 Stephen Lack. A 2-categories companion. In *Towards higher categories*, pages 105–191. Springer, 2010.

28 Stephen Lack and Ross Street. The formal theory of monads II. *Journal of pure and applied algebra*, 175(1-3):243–265, 2002.

29 Tom Leinster. Basic Bicategories, 1998. arXiv:math/9810017.

30 Paul Blain Levy. *Call-by-push-value: A Functional/imperative Synthesis*, volume 2. Springer Science & Business Media, 2012.

31 Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

32 Saunders MacLane and Robert Paré. Coherence for bicategories and indexed categories. *Journal of Pure and Applied Algebra*, 37:59–80, 1985.

33 The mathlib Community. The Lean mathematical library. In Jasmin Blanchette and Catalin Hritcu, editors, *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2020, New Orleans, LA, USA, January 20-21, 2020*, pages 367–381. ACM, 2020. doi:10.1145/3372885.3373824.

34 Paul-André Mellies. Categorical semantics of linear logic. *Panoramas et syntheses*, 27:15–215, 2009.

35 Eugenio Moggi. Computational lambda-calculus and monads. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*, pages 14–23. IEEE Computer Society, 1989. doi:10.1109/LICS.1989.39155.

36 Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991.

37 Tobias Nipkow, Lawrence C Paulson, and Markus Wenzel. *Isabelle/HOL: a proof assistant for higher-order logic*, volume 2283. Springer Science & Business Media, 2002.

38 Ulf Norell. Dependently typed programming in Agda. In *Proceedings of the 4th international workshop on Types in language design and implementation*, pages 1–2, 2009.

39 Gordon Plotkin and John Power. Notions of computation determine monads. In *International Conference on Foundations of Software Science and Computation Structures*, pages 342–356. Springer, 2002.

40 Gordon Plotkin and John Power. Algebraic operations and generic effects. *Applied categorical structures*, 11(1):69–94, 2003.

41 John Power. Coherence for bicategories with finite bilimits I. In *Categories in computer science and logic*, volume 92 of *Contemporary Mathematics*, pages 341–347, 1989.

42 John Power. Enriched Lawvere theories. *Theory and Applications of Categories*, 6(7):83–93, 1999.

43 John Power and Edmund Robinson. A Characterization of PIE Limits. *Mathematical Proceedings of the Cambridge Philosophical Society*, 110:33–47, 1991.

44 John Power and Hiroshi Watanabe. Combining a monad and a comonad. *Theoretical Computer Science*, 280(1-2):137–162, 2002.

45 Egbert Rijke, Elisabeth Bonnevier, Jonathan Prieto-Cubides, et al. Univalent mathematics in Agda. https://unimath.github.io/agda-unimath/. URL: https://github.com/UniMath/agda-unimath/.

46 Eugene W. Stark. Bicategories. *Arch. Formal Proofs*, 2020, 2020. URL: https://www.isa-afp.org/entries/Bicategory.html.

47 Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972.

48 The Coq Development Team. The Coq Proof Assistant, September 2022. doi:10.5281/zenodo.7313584.

49 The 1Lab Development Team. The 1Lab. URL: https://1lab.dev.

50 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

51 Niccolò Veltri and Niels van der Weide. Constructing Higher Inductive Types as Groupoid Quotients. *Logical Methods in Computer Science*, Volume 17, Issue 2, April 2021. URL: https://lmcs.episciences.org/7391.

52 Andrea Vezzosi, Anders Mörtberg, and Andreas Abel. Cubical Agda: a dependently typed programming language with univalence and higher inductive types. *Proc. ACM Program. Lang.*, 3(ICFP):87:1–87:29, 2019. doi:10.1145/3341691.

53 Vladimir Voevodsky, Benedikt Ahrens, Daniel Grayson, et al. Unimath — a computer-checked library of univalent mathematics. available at http://unimath.org. doi:10.5281/zenodo.7848572.

54 Kobe Wullaert, Ralph Matthes, and Benedikt Ahrens. Univalent monoidal categories. *arXiv preprint arXiv:2212.03146*, 2022.