The Formal Theory of Monads, Univalently

Niels van der Weide

3 July, 2023

Univalent Foundations

- Key aspect of **univalent foundations**: the univalence axiom
- **The univalence axiom**: isomorphism of types is the same as equality of types
- The foundations of libraries like **UniMath**[1].

[1] `https://github.com/UniMath/UniMath`

Category Theory in Univalent Foundations

- In univalent foundations, we are interested in **univalent categories**
- These are categories in which isomorphism between objects is the same as equality between them (*compare to the univalence axiom*)
- Semantically, this is the “right” notion.
- In addition, it is more convenient to work with univalent categories.

Overall Goal

This talk from a broader perspective:

- Develop category theory in univalent foundations
- Formalize it in a proof assistant
- Ultimately: also formalize applications of category theory (i.e., in logic or programming language theory)

Monads

Monads are one of the key concept in category theory. A monad on a category C is given by

- a functor $M : \mathsf{C} \to \mathsf{C}$
- a natural transformation $\eta : \mathbf{id} \Rightarrow M$ (the **unit**)
- a natural transformation $\mu : M \cdot M \Rightarrow M$ (the **multiplication**)

such that certain laws hold.

Monads

Monads are one of the key concept in category theory. A monad on a category C is given by

- a functor $M : \mathsf{C} \to \mathsf{C}$
- a natural transformation $\eta : \mathbf{id} \Rightarrow M$ (the **unit**)
- a natural transformation $\mu : M \cdot M \Rightarrow M$ (the **multiplication**)

such that certain laws hold.

Note: compare to monads in Haskell

- we have a type $m\ a$
- we have return : $a \to m\ a$
- we have $(>>=) : m\ a \to (a \to\ m\ b) \to m\ b$

Applications of Monads

- Monads are important in the study of programming languages.
- More specifically, monads can be used to study computational effects

Concrete applications of monads:

- Moggi's computational lambda calculus
- The enriched effect calculus
- Call-by-push-value
- Linear logic

The Theory of Monads

Key theorems about monads:

- every monad gives rise to an adjunction
- every adjunction gives rise to a monad

$$\begin{tikzcd} C \arrow[r, bend left, "F"] & E \arrow[l, bend left, "U"] \end{tikzcd}$$

The Theory of Monads

Key theorems about monads:

- every monad gives rise to an adjunction
- every adjunction gives rise to a monad

$$\begin{tikzcd} C \arrow[r, bend left, "F"] & E \arrow[l, bend left, "U"] \end{tikzcd}$$

There are two ways to obtain an adjunction from a monad

- Via Eilenberg-Moore categories
- Via Kleisli categories

Kleisli categories

Let M be a monad on C.
We define the Kleisli category of M as follows

- Objects: objects of C
- Morphisms from x to y in the Kleisli category are morphisms from x to $M\ y$ in C.

Problem!

- Recall that in univalent foundations, we are interested in **univalent categories** (*categories in which isomorphism is the same as identity*)
- The Kleisli category, as defined on the previous slide, is **not** univalent in general.
- A solution has been proposed, but the relevant theorems for it were not proven[2]

[2] Ahrens, Benedikt, Paige Randall North, Michael Shulman, and Dimitris Tsementzis. "*The univalence principle*."

Problem!

- We are also interested in formalization.
- Monads occur in many different flavors (e.g., enriched monads, monoidal monads, comonads)
- We don't want to reprove the relevant theorem for every kind of monad
- The formal theory of monads by Street gives a general framework for monads[3]

[3] Street, Ross. " *The formal theory of monads*."

Goals of the Paper

- In the paper, we develop the formal theory of monads in univalent foundations
- We instantiate this theory to various examples
- The results in the paper are formalized in Coq using the UniMath library.

Ingredients for the Formal Theory of Monad

There are three key ingredients in the formal theory of monads:

- Bicategories
- The bicategory of monads
- Eilenberg-Moore objects

Bicategories

- **Bicategories** give an abstract setting in which one can study category theory
- Many categorical notions have a bicategorical analogue
- We have bicategories of categories, of monoidal categories, and of enriched categories.

Category Theory	Bicategorical notion
Category	Object
Functor	1-cell
Natural transformation	2-cell
Adjunction	Internal adjunction

Bicategories vs 2-categories

Difference between bicategories and 2-categories:

- In a 2-category: for composable 1-cells f, g, h, we have $f \cdot (g \cdot h) = (f \cdot g) \cdot h$
- In bicategory: for composable 1-cells f, g, h, we have an isomorphism between $f \cdot (g \cdot h)$ and $(f \cdot g) \cdot h$

Bicategories vs 2-categories

Difference between bicategories and 2-categories:

- In a 2-category: for composable 1-cells f, g, h, we have $f \cdot (g \cdot h) = (f \cdot g) \cdot h$
- In bicategory: for composable 1-cells f, g, h, we have an isomorphism between $f \cdot (g \cdot h)$ and $(f \cdot g) \cdot h$

Note:

- As a result, the definition of a bicategory becomes more complicated.
- This is because we need to require **coherences** to get a well-behaved notion.

Bicategories vs 2-categories

Why we use bicategories:

- We work in **univalent foundations**, which is **intensional**.

Bicategories vs 2-categories

Why we use bicategories:

- We work in **univalent foundations**, which is **intensional**.
- In intensional foundations, there is a difference between **propositional equality** ($=$) and **definitional equality** ($\equiv$).
- We can replace equals by equals.

Bicategories vs 2-categories

Why we use bicategories:

- We work in **univalent foundations**, which is **intensional**.
- In intensional foundations, there is a difference between **propositional equality** ($=$) and **definitional equality** ($\equiv$).
- We can replace equals by equals.
- If $f \equiv g$: then in any term, we can replace f by g.
- If $P : f = g$: then we need to use **transport**.
- So: for $=$, the proof of equality is written in the term.

Bicategories vs 2-categories

Why we use bicategories:

- We work in **univalent foundations**, which is **intensional**.
- In intensional foundations, there is a difference between **propositional equality** ($=$) and **definitional equality** ($\equiv$).
- We can replace equals by equals.
- If $f \equiv g$: then in any term, we can replace f by g.
- If $P : f = g$: then we need to use **transport**.
- So: for $=$, the proof of equality is written in the term.
- The laws of a 2-category are **propositional equalities**.
- As such, if we use any law, then it will be present in the term.

Bicategories vs 2-categories

Why we use bicategories:

- We work in **univalent foundations**, which is **intensional**.
- In intensional foundations, there is a difference between **propositional equality** ($=$) and **definitional equality** ($\equiv$).
- We can replace equals by equals.
- If $f \equiv g$: then in any term, we can replace f by g.
- If $P : f = g$: then we need to use **transport**.
- So: for $=$, the proof of equality is written in the term.
- The laws of a 2-category are **propositional equalities**.
- As such, if we use any law, then it will be present in the term.
- Note: this subtlety does **not** come up in extensional foundations (like ZFC)

The Bicategory of Monads

Given a bicategory B, a **monad** in B consists of

- an object x : B
- a 1-cell $m : x \to x$
- a 2-cell $\eta : \mathbf{id} \Rightarrow m$
- a 2-cell $m \cdot m \Rightarrow m$

such that the monad laws are satisfied.

The Bicategory of Monads

Given a bicategory B, a **monad** in B consists of

- an object $x : \mathsf{B}$
- a 1-cell $m : x \to x$
- a 2-cell $\eta : \mathbf{id} \Rightarrow m$
- a 2-cell $m \cdot m \Rightarrow m$

such that the monad laws are satisfied.
Note:

- We can also define the notion of a **morphism between monads** and of a **2-cell** between such morphisms.
- This gives a bicategory of monads internal to B

Displayed Bicategories

Problem:

- We want to prove that the bicategory of monads is univalent.
- However, a direct proof is complicated
- This is because monads are objects with a lot of structure, so their equality is complicated.

Displayed Bicategories

Problem:

- We want to prove that the bicategory of monads is univalent.
- However, a direct proof is complicated
- This is because monads are objects with a lot of structure, so their equality is complicated.

Solution:

- Break up the structure and build the bicategory of monads step by step
- Tool: displayed bicategories.

Displayed Bicategories

Basic idea of construction:

- We start with B
- First, we add a 1-cell $m : x \to x$ to the structure
- Afterwards, we add the unit $\mathbf{id} \Rightarrow m$ and the multiplication $m \cdot m \Rightarrow m$ to the structure.
- Then add the laws

Displayed Bicategories

Basic idea of construction:

- We start with B
- First, we add a 1-cell $m : x \to x$ to the structure
- Afterwards, we add the unit $\mathbf{id} \Rightarrow m$ and the multiplication $m \cdot m \Rightarrow m$ to the structure.
- Then add the laws

Explanation:

- Instead of defining the bicategory of monads in one go, it is defined in multiple steps
- In each step, one part of the structure is added
- This simplifies the proof that the bicategory is univalent, because we can consider each piece of structure separately

Eilenberg-Moore Objects

- We can define the notion of Eilenberg-Moore objects in arbitrary bicategories by stating a universal property
- Eilenberg-Moore objects are examples of **limits**

Eilenberg-Moore Objects

- We can define the notion of Eilenberg-Moore objects in arbitrary bicategories by stating a universal property
- Eilenberg-Moore objects are examples of **limits**
- Usually, limits are **unique up to isomorphism**
- However, assuming univalence, limits are **unique up to equality**
- Consequence: if some bicategory has Eilenberg-Moore objects (not necessarily chosen), then we can choose them **without** the axiom of choice.

For details: see the paper

Kleisli Categories

- Note: Eilenberg-Moore objects are defined in arbitrary bicategories
- Eilenberg-Moore objects in $\mathsf{Cat}^{\mathsf{op}}$: Kleisli categories
- We must define Kleisli category slightly differently: as a full subcategory of the Eilenberg-Moore category
- Proving the universal property: Rezk completion

The relevant theorems now follow from the general framework.

Summary: how does univalence affect the development?

Univalence affected the development in the following ways:

- We need to use bicategories instead of 2-categories (*note that this is already so in intensional foundations*)
- Displayed bicategories become convenient, and the bicategory of monads is defined in a different way
- Eilenberg-Moore objects are unique up to equality instead of only up to equivalence.

Other stuff in the formalization

The paper also discusses:

- Numerous examples
- Distributive laws and composition of monads
- Adjunctions arising from monads
- Monadic 1-cells

Takeaways from this talk

- **For category theorists**: univalent foundations is a nice setting for studying category theory
- **For type theorists**: to properly study category theory in univalent foundations, some new methods are needed (*Rezk completions*)
- **For formalizers**: the formal theory of monads provides the proper level of abstraction for formalizing monads
- **For programming language theorists**: more and more categorical tools for programming language semantics are formalized