Category Theory in UniMath

Niels van der Weide

Radboud University, The Netherlands

April 4, 2019

This talk

- What are univalent categories?
- How to construct univalent categories?

Note on terminology: during this talk, I use terminology from UniMath (different from HoTT book).

Categories in Univalent Foundations

Definition (Precategory)

A *precategory* $\mathcal{C}$ consists of

- A type $\mathcal{C}_0$ of *objects*;
- For $x, y : \mathcal{C}_0$ a type $\mathcal{C}_1(x, y)$ of *morphisms*;
- For $x : \mathcal{C}_0$ an *identity* morphism $\mathrm{id}_x : \mathcal{C}_1(x, x)$;
- For $x, y, z : \mathcal{C}_0$ and $f : \mathcal{C}_1(x, y)$ and $g : \mathcal{C}_1(y, z)$, a *composition* $f \cdot g : \mathcal{C}_1(x, z)$

such that

- $f \cdot \mathrm{id}_x = f$;
- $\mathrm{id}_y \cdot f = f$;
- $f \cdot (g \cdot h) = (f \cdot g) \cdot h$.

Categories in Univalent Foundations

- Equality is proof relevant in UF.
- Precategories can have 'higher' structure given by the paths.
- Eg, the 1-cells are morphisms, 2-cells are equalities between morphisms.
- For categories, we want this to collapse.

Categories in Univalent Foundations

Definition (Category)

A *category* $\mathcal{C}$ consists of

- A type $\mathcal{C}_0$ of *objects*;
- For $x, y : \mathcal{C}_0$ a **set** $\mathcal{C}_1(x, y)$ of *morphisms*;
- For $x : \mathcal{C}_0$ an *identity* morphism $\mathrm{id}_x : \mathcal{C}_1(x, x)$;
- For $x, y, z : \mathcal{C}_0$ and $f : \mathcal{C}_1(x, y)$ and $g : \mathcal{C}_1(y, z)$, a *composition* $f \cdot g : \mathcal{C}_1(x, z)$

such that

- $f \cdot \mathrm{id}_x = f$;
- $\mathrm{id}_y \cdot f = f$;
- $f \cdot (g \cdot h) = (f \cdot g) \cdot h$.

(Recall: a set is a type for which equality is proof irrelevant)

Examples of Categories

- The category **SET** of sets and functions
- The category of pointed sets and point preserving maps
- The category of monoids and homomorphisms

Towards Univalent Categories: Isomorphisms

Definition

A morphism $f : \mathcal{C}_1(x, y)$ is an *isomorphism* if the map $\lambda(g : \mathcal{C}_1(y, z)), f \cdot g$ is an equivalence for every $z : \mathcal{C}_0$.
We denote the type of isomorphisms from X to Y by $X \cong Y$.

Note:

- We can find inverses.
- Being an isomorphism is a proposition
- id_x is an isomorphism

In UniMath: `is_iso`

Towards Univalent Categories: Isomorphisms

Alternatively, we can define

Definition

A morphism $f : \mathcal{C}_1(x, y)$ is an *isomorphism* if we have $g : \mathcal{C}_1(y, x)$ such that $f \cdot g = \mathrm{id}_x$ and $g \cdot f = \mathrm{id}_y$.

Note that these definitions are equivalent for categories.

In UniMath: `z_iso`.

Univalent Categories

Definition (Univalence Axiom)

- For all types X, Y we have a map idtoeq $X\ Y : X = Y \to X \simeq Y$.
- UA: the map $X = Y \to X \simeq Y$ is an equivalence.

Definition (Univalent Categories)

Let $\mathcal{C}$ be a category.

- For all objects $x, y : \mathcal{C}_0$ we have a map $\mathbf{idtoiso}_{x,y} : x = y \to x \cong y$.
- A category $\mathcal{C}$ is *univalent* if for all $x, y : \mathcal{C}_0$ the map $\mathbf{idtoiso}_{x,y}$ is an equivalence.

What's so good about univalent categories?

- Nice properties: initial objects are unique (exercise)
- It's the "right" notion of category **in univalent foundations**.
- In the simplicial set interpretation, univalent categories correspond to actual categories.

SET is Univalent

To prove **SET** is univalent, we factor **idtoiso** as follows.

$$
\begin{tikzcd}
x = y \arrow[rr, "\mathbf{idtoiso}_{x,y}"] \arrow[dr, "\simeq"'] & & x \cong y \\
& x \simeq y \arrow[ur, "\simeq"'] &
\end{tikzcd}
$$

Hence, **idtoiso** is equal to an equivalence and thus an equivalence.

What about Monoids?

- Is monoids a univalent category?
- Monoids have a more complicated structure, which makes a direct proof harder.
- We need machinery to make such proofs more manageable.
- For this, we use *displayed categories*

Displayed Categories, The Idea

- Suppose, we have a category $\mathcal{C}$.
- A displayed category $\mathcal{D}$ represents "structure" or "properties" to be added to $\mathcal{C}$.
- Displayed categories give rise to a *total category* $\int \mathcal{D}$
- The objects of $\int \mathcal{D}$ are pairs of $x : \mathcal{C}_0$ with the extra structure.
- Furthermore, we have a projection (forgetful functor) from the total category to $\mathcal{C}$.
- Goal of displayed categories: reason about the total category.

Displayed Categories, The Data

Definition

A *displayed category* $\mathcal{D}$ over $\mathcal{C}$ consists of

- For each $x : \mathcal{C}_0$ a type $\mathcal{D}_0^x$ of *objects over* x.
- For each $f : \mathcal{C}_1(x, y)$, $\overline{x} : \mathcal{D}_0^x$ and $\overline{y} : \mathcal{D}_0^y$ a set $\mathcal{D}_1^f(\overline{x}, \overline{y})$ of *morphisms over* f.
- For each $x : \mathcal{C}_0$ and $\overline{x} : \mathcal{D}_0^x$ an identity $\overline{\mathrm{id}_x} : \mathcal{D}_1^{\mathrm{id}_x}(\overline{x}, \overline{x})$.
- For $f : \mathcal{C}_1(x, y)$, $g : \mathcal{C}_1(y, z)$, $\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$. and $\overline{g} : \mathcal{D}_1^g(\overline{y}, \overline{z})$, a composition $\overline{f} \cdot \overline{g} : \mathcal{D}_1^{f \cdot g}(\overline{x}, \overline{z})$.

What about the laws?

Displayed Categories, Towards The Laws

Let's try to write the right unitality law.
Suppose $\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$. Then

$$\overline{f} \cdot \overline{\mathsf{id}_y} : \mathcal{D}_1^{f \cdot \mathsf{id}_y}(\overline{x}, \overline{y})$$

Hence, the law $\overline{f} = \overline{f} \cdot \overline{\mathsf{id}_y}$ does *not* type check.

Displayed Categories, Towards The Laws

Let's try to write the right unitality law.
Suppose $\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$. Then

$$\overline{f} \cdot \overline{\mathsf{id}_y} : \mathcal{D}_1^{f \cdot \mathsf{id}_y}(\overline{x}, \overline{y})$$

Hence, the law $\overline{f} = \overline{f} \cdot \overline{\mathsf{id}_y}$ does *not* type check.
Solution: use transport. Laws become *dependent equalities*.

Displayed Categories, The Laws

Suppose, $f, g : \mathcal{C}_1(x, y)$ and $p : f = g$. Then

$$\mathsf{transport}^{\lambda h, \mathcal{D}_1^h(\overline{x}, \overline{y})}\ p : \mathcal{D}_1^f(\overline{x}, \overline{y}) \to \mathcal{D}_1^g(\overline{x}, \overline{y})$$

Recall that

$$\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$$

$$f \cdot \overline{\mathsf{id}_y} : \mathcal{D}_1^{f \cdot \mathsf{id}_y}(\overline{x}, \overline{y})$$

So, it suffices to find a path $f = f \cdot \mathsf{id}_y$.
This is one of the axioms of categories.

The Total Category

Definition

Let $\mathcal{D}$ be a displayed category over $\mathcal{C}$. Then we define the *total category* $\int \mathcal{D}$ to be the category for which

- objects are pairs $x : \mathcal{C}_0$ and $\overline{x} : \mathcal{D}_0^x$
- morphisms from $(x, \overline{x})$ to $(y, \overline{y})$ are pairs $f : \mathcal{C}_1(x, y)$ and $\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$

Definition

We have a *projection* functor $\pi_1 : \int D \longrightarrow C$. It sends $(x, \overline{x})$ to x and $(f, \overline{f})$ to f.

Examples of Displayed Categories: Pointed Sets

Define a displayed category P over $\mathbf{SET}$:

- Objects over X are elements $x : X$
- Morphisms over $f : X \to Y$ from $x : X$ to $y : Y$ are paths $f\ x = y$
- Morphism over id_X is a path $\mathrm{id}_X\ x = x$ (reflexivity)

The total category $\int P$ is the category of *pointed sets*.
Objects: pair of a set X and $x : X$. Morphisms: point preserving maps.

Examples of Displayed Categories: Monoids

Define a displayed category over **SET**

- Objects over X are monoid structures
- Morphisms over f are proofs that f is a homomorphism

The total category is the category of monoids.

Constructions with Displayed Categories

Some constructions which allow building displayed categories modularly.

- The full subcategory is a displayed category
- We can take the product of displayed categories

A Nicer Construction of the Category of Monoids

Note: displayed categories can be layered.

- Start with the category of sets.
- Define a displayed category P on sets. Objects over X are points.

A Nicer Construction of the Category of Monoids

Note: displayed categories can be layered.

- Start with the category of sets.
- Define a displayed category P on sets. Objects over X are points.
- Define a displayed category M' on sets. Objects over X are maps $X \times X \to X$.

A Nicer Construction of the Category of Monoids

Note: displayed categories can be layered.

- Start with the category of sets.
- Define a displayed category P on sets. Objects over X are points.
- Define a displayed category M' on sets. Objects over X are maps $X \times X \to X$.
- This gives a displayed category $P \times M'$ over sets (the product)
- Call its total category $\mathcal{E}$.
- Objects of $\mathcal{E}$ are pairs $(X, (e, f))$ with $e : X$ and $f : X \to X$

A Nicer Construction of the Category of Monoids

Note: displayed categories can be layered.

- Start with the category of sets.
- Define a displayed category P on sets. Objects over X are points.
- Define a displayed category M' on sets. Objects over X are maps $X \times X \to X$.
- This gives a displayed category $P \times M'$ over sets (the product)
- Call its total category $\mathcal{E}$.
- Objects of $\mathcal{E}$ are pairs $(X, (e, f))$ with $e : X$ and $f : X \to X$
- Define a displayed category M over $\mathcal{E}$. Objects over $(X, (e, f))$ are proofs that it's a monoid.
- Then the total category of M is the category of monoids.

A Nicer Construction of the Category of Monoids

Note: displayed categories can be layered.

- Start with the category of sets.
- Define a displayed category P on sets. Objects over X are points.
- Define a displayed category M' on sets. Objects over X are maps $X \times X \to X$.
- This gives a displayed category $P \times M'$ over sets (the product)
- Call its total category $\mathcal{E}$.
- Objects of $\mathcal{E}$ are pairs $(X, (e, f))$ with $e : X$ and $f : X \to X$
- Define a displayed category M over $\mathcal{E}$. Objects over $(X, (e, f))$ are proofs that it's a monoid.
- Then the total category of M is the category of monoids.

Untangling (break down in small parts) and *stratification* (layers)

Towards Displayed Univalence: Displayed Isomorphisms

Definition

Let $\mathcal{D}$ be a displayed category over $\mathcal{C}$ and suppose, f is an isomorphism with inverse g. We say $\overline{f} : \mathcal{D}_1^f(\overline{x}, \overline{y})$ is a *(displayed) isomorphism* if there is $\overline{g} : \mathcal{D}_1^g(\overline{y}, \overline{x})$ which are mutual inverses (again as dependent equalities).
We write $\overline{x} \cong_f \overline{y}$ for the type of displayed isomorphisms over f.

Displayed Univalence

- Again the identity $\overline{\mathrm{id}_x}$ is an isomorphism
- By path induction, we get for each $p : x = y$ a map

$$\mathbf{dispidtoiso}_{\overline{x},\overline{y}} : \overline{x} =_p \overline{y} \to \overline{x} \cong_{\mathbf{idtoiso}_{x,y}\, p} \overline{y}$$

- We say $\mathcal{D}$ is *displayed univalent* if **dispidtoiso** is an equivalence.

Main Theorem

Theorem

If $\mathcal{C}$ is univalent and $\mathcal{D}$ is displayed univalent, then $\int \mathcal{D}$ is univalent.

Examples of Displayed Univalent Categories

- The displayed category P of pointed sets is displayed univalent and thus the category of pointed sets is univalent.
- The displayed category M of monoids is displayed univalent and thus the category of monoids is univalent.

Conclusion

Take away message:

- Displayed categories are a convenient way to modularly construct univalent categories.
- Work with small “edible” pieces.

In the exercises:

- Study univalent categories more closely
- Define monoids as a displayed category

Literature

- HoTT Book, Chapter 9
- Ahrens, Benedikt and Lumsdaine, Peter LeFanu. "Displayed Categories." Logical Methods in Computer Science 15 (2019).
- Ahrens, B., Kapulkin, K., & Shulman, M. (2015). Univalent Categories and the Rezk Completion. Mathematical Structures in Computer Science, 25(5), 1010-1039.