Formalizing Double Categories in UniMath

Niels van der Weide, j.w.w. Nima Rasekh, Benedikt Ahrens,
Paige Randall North

This talk

This material of this talk is based on two papers

- "**Univalent Double Categories**" by N. van der Weide, N. Rasekh, B. Ahrens, P.R. North[1]
- "**Insights From Univalent Foundations: A Case Study Using Double Categories**" by N. Rasekh, N. van der Weide, B. Ahrens, P.R. North[2]

Our focus is on the first of these papers

Slides are available at:

`https://nmvdw.github.io/pubs/seminar-coreact.pdf`

[1] `https://doi.org/10.1145/3636501.3636955`

[2] `https://doi.org/10.48550/arXiv.2402.05265`

Brief Introduction to Double Categories

Approaches to Formalizing Double Categories

2-Sided Displayed Categories

Univalence

Conclusion

What are Double Categories?

A **double category** is given by:

- **objects**
- **vertical morphisms**
- **horizontal morphisms**
- and **squares**

with suitable composition and identity operations

$$
\begin{tikzcd}
 & sq \arrow[dl, shift left] \arrow[dl, shift right] \arrow[dr, shift left] \arrow[dr, shift right] & \\
hor \arrow[dr, shift left] \arrow[dr, shift right] & & ver \arrow[dl, shift left] \arrow[dl, shift right] \\
 & ob &
\end{tikzcd}
$$

Strictness

Double categories come with various notions of strictness:

- Strict in both directions: **strict double categories**
- Weak in one direction: **pseudo double categories**
- Weak in both directions: **weak double categories**

Examples

	Objects	Horizontal	Vertical	Strictness
Rel	sets	functions	relations	strict
Span($\mathcal{C}$)	objects in $\mathcal{C}$	morphisms in $\mathcal{C}$	spans in $\mathcal{C}$	pseudo
Prof	categories	functors	profunctors	pseudo
Sq($\mathcal{B}$)	objects in $\mathcal{B}$	1-cells in $\mathcal{B}$	1-cells in $\mathcal{B}$	weak

Here $\mathcal{C}$ is a category with pullbacks and $\mathcal{B}$ is a bicategory
Note: **strict double categories** are rare in practice, while the **weaker** versions are more common

Brief Introduction to Double Categories

Approaches to Formalizing Double Categories

2-Sided Displayed Categories

Univalence

Conclusion

Formalizing Double Categories

Our requirements for formalizations of double categories are:

- it includes **pseudo double categories** and not only strict ones
- the notion of double category should be built on **reusable/modular** definitions

Formalizing Double Categories

Our requirements for formalizations of double categories are:

- it includes **pseudo double categories** and not only strict ones
- the notion of double category should be built on **reusable/modular** definitions

In the papers, we also look at

- **univalence** (because we work in univalent foundations)
- weak double categories

However, neither of these are the focus of this talk.

Weakness and Intensionality

Most proof assistants are based on intensional foundations, which further decreases the use of strict structures.

- The strength of strictness in extensional foundations comes from the fact that we can remove associators/unitors from diagrams
- So: if $\tau : k \Rightarrow f \cdot (g \cdot h)$ and $\theta : (f \cdot g) \cdot h \Rightarrow k$, we can write $\tau \cdot \theta$
- However, in intensional foundations, we can only assume a provable equality $f \cdot (g \cdot h) = (f \cdot g) \cdot h$
- Result: we cannot write $\tau \cdot \theta$, but we have to include the relevant equalities

So: pseudo double categories are a **must**

Common Approach: Internal Categories

Usually, double categories are defined using **internal categories**.
So, a double category is given by functors $s, t : \mathcal{C}_1 \to \mathcal{C}_0$ as follows:

$$\begin{tikzcd} \mathcal{C}_1 \arrow[d, shift right, "s"'] \arrow[d, shift left, "t"] \\ \mathcal{C}_0 \end{tikzcd}$$

such that we have suitable identity and composition functors.
However, we chose **not** to do so.

Composition and Pullbacks

To express composition, we must have a **composable** pair of arrows

$$\begin{tikzcd} \mathcal{C}_1 \arrow[d, shift right, "s"'] \arrow[d, shift left, "t"] \\ \mathcal{C}_0 \end{tikzcd}$$

f g

$$x = s(f) \to t(f) = y = s(g) \to t(g) = z$$

Composition and Pullbacks

So, we take the following pullback

$$\begin{tikzcd}
& & P \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\lrcorner" rotate=-45, very near start] & & \\
& \mathcal{C}_1 \arrow[dl, "s"'] \arrow[dr, "t"] & & \mathcal{C}_1 \arrow[dl, "s"'] \arrow[dr] & \\
\mathcal{C}_0 & & \mathcal{C}_0 & & \mathcal{C}_1
\end{tikzcd}$$

Associativity and Scary Pullbacks

For associativity, it becomes messier

$$
\begin{tikzcd}
 & & & R \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\vee"] & & & \\
 & & P \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\vee"] & & Q \arrow[dl] \arrow[dr] \arrow[dd, phantom, "\vee"] & & \\
 & \mathcal{C}_1 \arrow[dl, "s"'] \arrow[dr, "t"] & & \mathcal{C}_1 \arrow[dl, "s"'] \arrow[dr, "t"] & & \mathcal{C}_1 \arrow[dl, "s"'] \arrow[dr, "t"] & \\
\mathcal{C}_0 & & \mathcal{C}_0 & & \mathcal{C}_0 & & \mathcal{C}_0
\end{tikzcd}
$$

Internal Categories: why not?

We chose not to use internal categories, because

- the pullbacks are complicated to handle
- an internal category in the 1-category of categories is a strict double category
- to get pseudo double categories, we must use pseudocategories internal to a 2-category

Category Theory and Dependent Types

In dependent type theory, we can define categories in 2 ways:

1. a type O of objects, and for all $x, y : O$ a type $M(x, y)$ of morphisms
2. a type O of objects, a type M of morphisms, and functions $s, t : M \to O$

Category Theory and Dependent Types

In dependent type theory, we can define categories in 2 ways:

1. a type O of objects, and for all $x, y : O$ a type $M(x, y)$ of morphisms
2. a type O of objects, a type M of morphisms, and functions $s, t : M \to O$

The first approach is nicer

- it is used more often in formalizations with dependent type theory:
- one can express composable pairs of arrows very directly
- the language is closer to how we do category theory in practice

This suggests another approach to defining double categories: an **unfolded definition**

An Unfolded Definition of Double Categories

We can define double categories in an unfolded way.
A double category $\mathcal{C}$ consists of:

- a category $\mathcal{C}_0$ of **objects** and **vertical morphisms**;
- for all objects $x, y : \mathcal{C}_0$ a type $x \rightarrow_h y$ of **horizontal morphisms**;
- for all objects $x : \mathcal{C}_0$ an **identity morphism** $i : x \rightarrow_h x$;
- for all morphisms $h : x \rightarrow_h y$ and $k : y \rightarrow_h z$ a **composition** $f \cdot g : x \rightarrow_h z$;
- for all vertical morphisms $v_1 : w \rightarrow x$ and $v_2 : y \rightarrow z$ and horizontal morphisms $h_1 : w \rightarrow_h y$ and $h_2 : x \rightarrow_h z$, a type of **squares** with sides v_1, v_2, h_1, and h_2
- and so on...

An Unfolded Definition: why not?

Advantages of the unfolded definition:

- we can express composable pairs directly
- there is no need for pullbacks, so the definition becomes simpler

However, this definition is not modular.

- it does not reuse many notions
- it does not consist of reusable parts

Our Approach

Our approach is based on **2-sided displayed categories**

- In style, it is close to the **unfolded definition**
- However, we identified reusable parts in the definition
- We split the unfolded definition into these reusable parts

The result is a modular definition of double categories avoiding pullbacks

Brief Introduction to Double Categories

Approaches to Formalizing Double Categories

2-Sided Displayed Categories

Univalence

Conclusion

A Common Pattern: Spans

Spans of categories are ubiquitous

- Double categories are spans

$$C_0 \xleftarrow{s} C_1 \xrightarrow{t} C_0$$

- A profunctor P from A to B can be represented as a span:

$$A \xleftarrow{\pi_1} \mathsf{Graph}(P) \xrightarrow{\pi_2} B$$

- The comma category of $F : \mathcal{C} \to \mathcal{E}$ and $G : \mathcal{D} \to \mathcal{E}$ forms a span:

$$\mathcal{C} \xleftarrow{\pi_1} \mathsf{Comma}(F, G) \xrightarrow{\pi_2} \mathcal{D}$$

A Common Pattern: Spans

Spans of categories are ubiquitous

- Double categories are spans

$$C_0 \xleftarrow{s} C_1 \xrightarrow{t} C_0$$

- A profunctor P from A to B can be represented as a span:

$$A \xleftarrow{\pi_1} \mathrm{Graph}(P) \xrightarrow{\pi_2} B$$

- The comma category of $F : \mathcal{C} \to \mathcal{E}$ and $G : \mathcal{D} \to \mathcal{E}$ forms a span:

$$\mathcal{C} \xleftarrow{\pi_1} \mathrm{Comma}(F, G) \xrightarrow{\pi_2} \mathcal{D}$$

2-sided displayed categories are an alternative presentation of spans.

2-sided displayed categories

double categories

2-sided displayed categories

profunctors

categorical constructions

Intuition from functions

There are two ways to represent maps from A to B:

1. via the function type $f : A \to B$
2. via the fibers $B \to \mathrm{Type}$ $(\lambda(b : B).f^{-1}(b))$

Note: every $P : B \to \mathrm{Type}$ gives rise to

- a type $A = \sum(b : B), P(b)$
- a function $f : A \to B$ sending x to $\pi_1(x)$

Intuition from functions

There are two ways to represent maps from A to B:

1. via the function type $f : A \to B$
2. via the fibers $B \to \text{Type}\ (\lambda(b : B).f^{-1}(b))$

Note: every $P : B \to \text{Type}$ gives rise to

- a type $A = \sum(b : B), P(b)$
- a function $f : A \to B$ sending x to $\pi_1(x)$

While functors use the first style (via functions), **displayed categories** and **2-sided displayed categories** use the second style (fiberwise)

What are displayed categories?

Definition

Let $\mathcal{C}$ be a category. A **displayed category**[3] $\mathcal{D}$ over $\mathcal{C}$ consists of

- for all $x : \mathcal{C}$ a type $\mathcal{D}_x$ of objects over x
- for all morphisms $f : x_1 \to x_2$ in $\mathcal{C}$, and objects $z_1 : \mathcal{D}_{x_1}$ and $z_2 : \mathcal{D}_{x_2}$, a type $z_1 \to_f z_2$ of morphisms over f and g
- for all objects $x : \mathcal{C}$ and $z : \mathcal{D}$ over x, an identity morphism $\overline{\mathrm{id}} : z \to_{\mathrm{id}} z$
- for all $h : z_1 \to_{f_1} z_2$ and $k : z_2 \to_{f_2} z_3$, a composition $h \cdot k : z_1 \to_{f_1 \cdot f_2} z_3$

The laws hold as dependent equalities over the corresponding laws in $\mathcal{C}$.

[3] Ahrens, Benedikt, and Peter LeFanu Lumsdaine. "*Displayed categories*."

What are displayed categories?

Definition

Let $\mathcal{C}$ be a category. A **displayed category**[3] $\mathcal{D}$ over $\mathcal{C}$ consists of

- for all $x : \mathcal{C}$ a type $\mathcal{D}_x$ of objects over x
- for all morphisms $f : x_1 \to x_2$ in $\mathcal{C}$, and objects $z_1 : \mathcal{D}_{x_1}$ and $z_2 : \mathcal{D}_{x_2}$, a type $z_1 \to_f z_2$ of morphisms over f and g
- for all objects $x : \mathcal{C}$ and $z : \mathcal{D}$ over x, an identity morphism $\overline{\mathsf{id}} : z \to_{\mathsf{id}} z$
- for all $h : z_1 \to_{f_1} z_2$ and $k : z_2 \to_{f_2} z_3$, a composition $h \cdot k : z_1 \to_{f_1 \cdot f_2} z_3$

The laws hold as dependent equalities over the corresponding laws in $\mathcal{C}$.

This is some kind of **dependent category**

[3] Ahrens, Benedikt, and Peter LeFanu Lumsdaine. "*Displayed categories*."

Laws of Displayed Categories

Suppose, we have

- a morphisms $f : x_1 \to x_2$ in $\mathcal{C}$
- objects $z_1 : \mathcal{D}_{x_1}$ and $z_2 : \mathcal{D}_{x_2}$
- a morphism $h : z_1 \to_f z_2$

Then we have $\overline{\mathrm{id}} \cdot h : z_1 \to_{\mathrm{id} \cdot f} z_2$

Laws of Displayed Categories

Suppose, we have

- a morphisms $f : x_1 \to x_2$ in $\mathcal{C}$
- objects $z_1 : \mathcal{D}_{x_1}$ and $z_2 : \mathcal{D}_{x_2}$
- a morphism $h : z_1 \to_f z_2$

Then we have $\overline{\mathsf{id}} \cdot h : z_1 \to_{\mathsf{id} \cdot f} z_2$
Not the same type as h, so we cannot write $\overline{\mathsf{id}} \cdot h = h$

Laws of Displayed Categories

Suppose, we have

- a morphisms $f : x_1 \to x_2$ in $\mathcal{C}$
- objects $z_1 : \mathcal{D}_{x_1}$ and $z_2 : \mathcal{D}_{x_2}$
- a morphism $h : z_1 \to_f z_2$

Then we have $\overline{\mathrm{id}} \cdot h : z_1 \to_{\mathrm{id} \cdot f} z_2$
Not the same type as h, so we cannot write $\overline{\mathrm{id}} \cdot h = h$
Since we have $p : \mathrm{id} \cdot f = f$, we can use a dependent equality, i.e. $\overline{\mathrm{id}} \cdot h =_p h$.

What are 2-sided displayed categories?

Definition

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be categories. A **2-sided displayed category** $\mathcal{D}$ over $\mathcal{C}_1$ and $\mathcal{C}_2$ consists of

- for all $x : \mathcal{C}_1$ and $y : \mathcal{C}_2$ a type $\mathcal{D}_{x,y}$ of objects over x and y
- for all morphisms $f : x_1 \to x_2$ in $\mathcal{C}_1$ and $g : y_1 \to y_2$ in $\mathcal{C}_2$, and objects $z_1 : \mathcal{D}_{x_1,y_1}$ and $z_2 : \mathcal{D}_{x_2,y_2}$, a type $z_1 \to_{f,g} z_2$ of morphisms over f and g
- for all objects $x : \mathcal{C}_1$ and $y : \mathcal{C}_2$ and objects $z : \mathcal{D}_{x,y}$ over x and y, an identity morphism $z \to_{\mathrm{id},\mathrm{id}} z$
- for all $h : z_1 \to_{f_1,g_1} z_2$ and $k : z_2 \to_{f_2,g_2} z_3$, a composition $h \cdot k : z_1 \to_{f_1 \cdot f_2, g_1 \cdot g_2} z_3$

The laws hold as dependent equalities over the corresponding laws in $\mathcal{C}_1$ and $\mathcal{C}_2$.

Example: Arrows

Example

Let $\mathcal{C}$ be a category. Define $\mathrm{arr}(\mathcal{C})$ as follows:

- objects over x and y: morphisms $x \to y$
- morphisms over $f : x_1 \to x_2$ and $g : y_1 \to y_2$ from $h_1 : x_1 \to y_1$ to $h_2 : x_2 \to y_2$: commuting squares

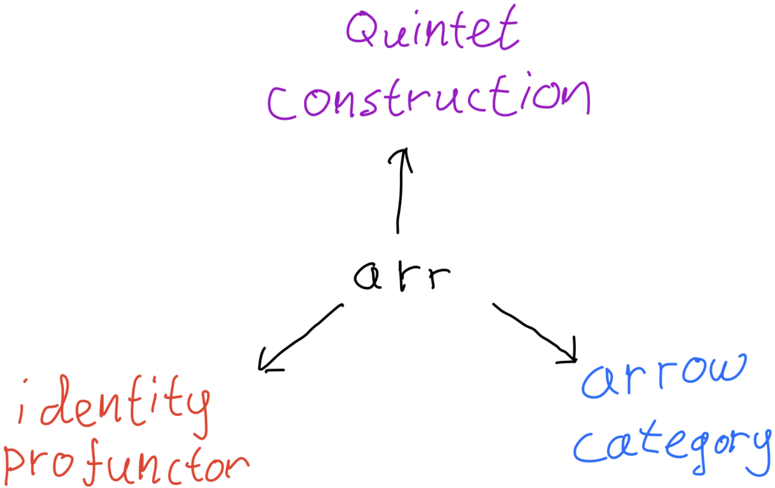
The Total Category

Every 2-sided displayed category $\mathcal{D}$ over $\mathcal{C}_1$ and $\mathcal{C}_2$ gives rise to a **total category** $\int \mathcal{D}$:

- Objects: triples $x : \mathcal{C}_1$, $y : \mathcal{C}_2$ and $z : \mathcal{D}_{x,y}$
- Morphsisms from (x_1, y_1, z_1) to (x_2, y_2, z_2): triples $f : x_1 \to x_2$, $g : y_1 \to y_2$ and $h : z_1 \to_{f,g} z_2$

In addition, there are projections to $\mathcal{C}_1$ and $\mathcal{C}_2$ taking the first and second components.

Arrows


Quintet construction
arr
identity profunctor
arrow category

2-sided displayed categories and spans

2-sided displayed category

$$\begin{tikzcd}
& z & \\
& \mathcal{D} & \\
\mathcal{C}_1 & & \mathcal{C}_2 \\
x & & y
\end{tikzcd}$$

spans

$$\begin{tikzcd}
& (x,y,z) & \\
& \int \mathcal{D} \arrow[dl] \arrow[dr] & \\
\mathcal{C}_1 & & \mathcal{C}_2 \\
x & & y
\end{tikzcd}$$

2-sided displayed categories and displayed categories

2-sided displayed category

$$\begin{tikzcd} & \overset{z}{\mathcal{D}} & \\ \underset{x}{\mathcal{C}_1} & & \underset{y}{\mathcal{C}_2} \end{tikzcd}$$

spans

$$\begin{tikzcd} & \overset{(x,y,z)}{\int \mathcal{D}} \arrow[dl] \arrow[dr] & \\ \underset{x}{\mathcal{C}_1} & & \underset{y}{\mathcal{C}_2} \end{tikzcd}$$

$$\overset{z}{\mathcal{D}}$$

$$\underset{(x,y)}{\mathcal{C}_1 \times \mathcal{C}_2}$$

displayed category

$$\begin{tikzcd} \overset{((x,y),z)}{\int \mathcal{D}} \arrow[d, shift left] \arrow[d, shift right] \\ \underset{(x,y)}{\mathcal{C}_1 \times \mathcal{C}_2} \end{tikzcd}$$

functor

2-sided displayed categories and double categories

A category $\mathcal{C}$ together with a 2-sided displayed category $\mathcal{D}$ over $\mathcal{C}$ and $\mathcal{C}$ gives us:

- a category $\mathcal{C}$ of **objects** and **vertical morphisms**
- the displayed objects of $\mathcal{D}$ represent **horizontal morphisms**
- the displayed morphisms of $\mathcal{D}$ represent **squares**

We also have **vertical identity squares** and and **vertical composition of squares**
What is missing:

- horizontal identity
- horizontal composition
- unitors, associators
- triangle and pentagon coherence

2-sided displayed categories and double categories

A double category is thus a 2-sided displayed category together with the following structure:

- horizontal identities
- horizontal composition
- unitors, associators
- triangle and pentagon coherence

These are done in an 'unfolded style'

2-sided displayed categories in category theory

2-sided displayed categories can be used to define

- **double categories**: require suitable composition and identity operations
- **profunctors**: require it to be a 2-sided discrete fibration

In addition, many constructions are instances of 2-sided displayed categories (arrow category, comma category, iso-comma category, spans, cospans, . . .)

Brief Introduction to Double Categories

Approaches to Formalizing Double Categories

2-Sided Displayed Categories

Univalence

Conclusion

Univalence

- UniMath uses **univalent foundations**
- As such, our work focuses on **univalent categories**
- The **univalence axiom** offers interesting persectives on category theory
- Univalence axiom: equality of types is the same as equivalence of types $((X \cong Y) \cong (X = Y))$.

In the remainder, I will highlight some aspects of univalence in our work.

Univalent Categories

Definition

A category $\mathcal{C}$ is univalent if for all objects x and y the types $x = y$ and $x \cong y$ are equivalent.

Example: by the univalent axiom, the category of sets is univalent.

Univalence and Category Theory

We also have a univalence principles for univalent categories: the types $\mathcal{C}_1 = \mathcal{C}_2$ is equivalent to the type of adjoint equivalences between $\mathcal{C}_1$ and $\mathcal{C}_2$.

- We get more powerful methods to handle equivalences of **univalent categories**. We can prove statements $\forall \mathcal{C}_1 \forall \mathcal{C}_2 \forall (e : \mathcal{C}_1 \cong \mathcal{C}_2), P(e)$ by **induction**: i.e., we can assume that e is the identity equivalence.
- Transport along equivalences holds automatically. Whenever $\mathcal{C}$ satisfies some property P and $\mathcal{C} \cong \mathcal{C}'$, then $\mathcal{C}'$ also satisfies P

Characterizing Equivalences using Univalence

Theorem

Every fully faithful and essentially surjective pseudo double functor is an adjoint equivalence.

Section 7 in "**Univalent Double Categories**": we prove this using equivalence induction

Main idea

We have forgetful pseudofunctors of bicategories

$$\begin{array}{c} \mathsf{DoubleCat} \\ \downarrow \\ \cdots \\ \downarrow \\ \mathsf{TwoSidedDispCat} \end{array}$$

Basically, we show that each forgetful functor reflects adjoint equivalence.
Technical ingredients: equivalence induction and displayed bicategories

The Univalence Maxim

We also have the **univalence maxim**.

- Double categories come with various notions of equivalence
- For each notion of equivalence, we have a suitable notion of univalent double category for which identity corresponds to the given notion of equivalence

This is the topic of: "**Insights From Univalent Foundations: A Case Study Using Double Categories**"

Brief Introduction to Double Categories

Approaches to Formalizing Double Categories

2-Sided Displayed Categories

Univalence

Conclusion

What do we have in UniMath[4]

Our formalization contains:

- strict double categories
- pseudo double categories via 2-sided displayed categories (and equivalence to unfolded definition)
- Verity double bicategories
- the bicategory of pseudo double categories, lax double functors and transformations
- basic theory of companion pairs and conjoints (on the level of Verity double bicategories)
- basic theory of gregarious equivalences (on the level of Verity double bicategories)
- the underlying 2-categories and bicategories of double categories

[4] `https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/DoubleCategories`

What do we have in UniMath

Our formalization also contains:

- characterization of equivalences and invertible 2-cells of pseudo double categories (**here we use univalence**)
- notions of univalence for pseudo double categories and Verity double bicategories
- univalence principles for strict double categories and for pseudo double categories

What do we have in UniMath

We got the following examples:

- Spans
- Structured cospans
- Relations
- Squares (for categories and for bicategories)
- Profunctors (both for univalent categories and strict categories)
- Transposes and opposites

Enriched profunctors is mostly done, but not completely finished.

Conclusion

- There are many ways to formalize double categories: internal categories, an unfolded definition, 2-sided displayed categories
- 2-sided displayed categories give a **modular** and **convenient** way to formalize double categories without pullbacks
- A 2-sided displayed category describes a span, and it is phrased in a more "dependently typed" style
- Univalence gives a more refined language for equivalences of (double) categories
- Univalence principles can help simplifying proofs about equivalences