Double Categories in Univalent Foundations

Nima Rasekh
joint with Niels van der Weide, Benedikt Ahrens, Paige North

Universität Greifswald

24.10.2024

Goal of the Talk

- Double categories come with many different types of *equivalences*.
- These equivalences influence our perspective on *double category theory*.
- *Set-theoretical* foundations cannot capture these equivalences by definition.
- *Univalent* foundations can mitigate the situation.

Double Categories throughout Mathematics

Double categories manifest in many different forms throughout mathematics. Our use cases naturally determine a relevant type of equivalences, which are not apparent from the definition.
Let's analyze a couple examples:

1. Double categories applied to 2-limits.
2. Double categories that arise as embeddings of 2-categories ...
   - ... with discrete vertical morphisms.
   - ... via the square embedding.

Underlying Horizontal Categories

Definition

Let $\mathbb{D}$ be a double category.

1. The **first underlying horizontal category** is the category given by:
   - Objects: Objects of $\mathbb{D}$
   - Morphisms: Horizontal morphisms of $\mathbb{D}$
2. The **second underlying horizontal category** is the category given by:
   - Objects: Vertical morphisms of $\mathbb{D}$
   - Morphisms: Squares of $\mathbb{D}$

Horizontally Terminal Objects

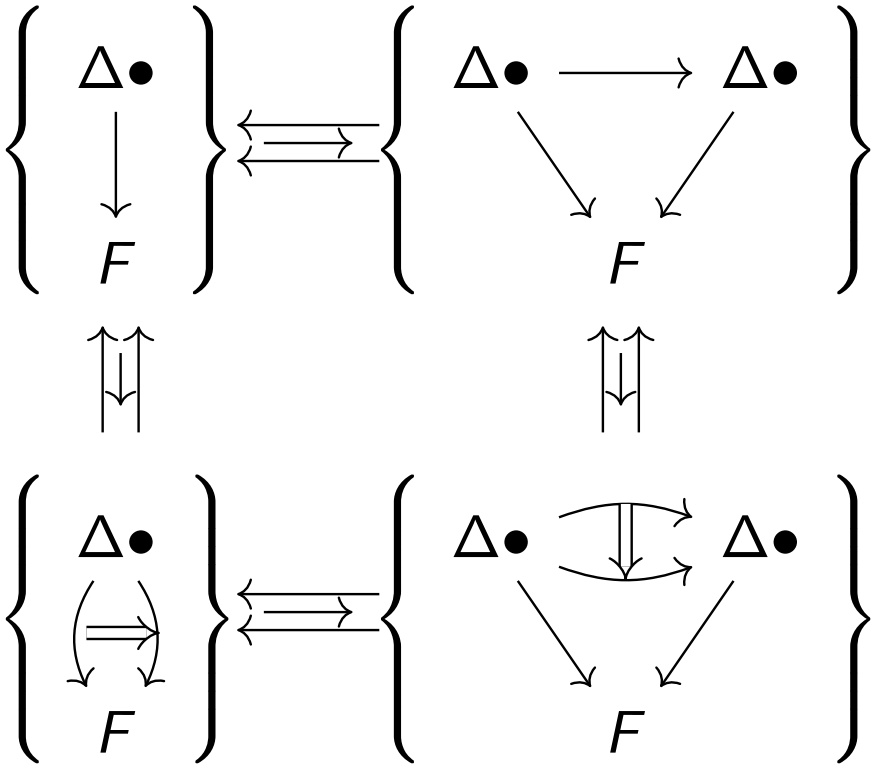
Definition

A double category $\mathbb{D}$ has a *horizontally terminal object* if there is an object t in $\mathbb{D}$, such that

1. t is terminal in the first underlying horizontal category.
2. The vertical identity on t is terminal in the second underlying horizontal category.

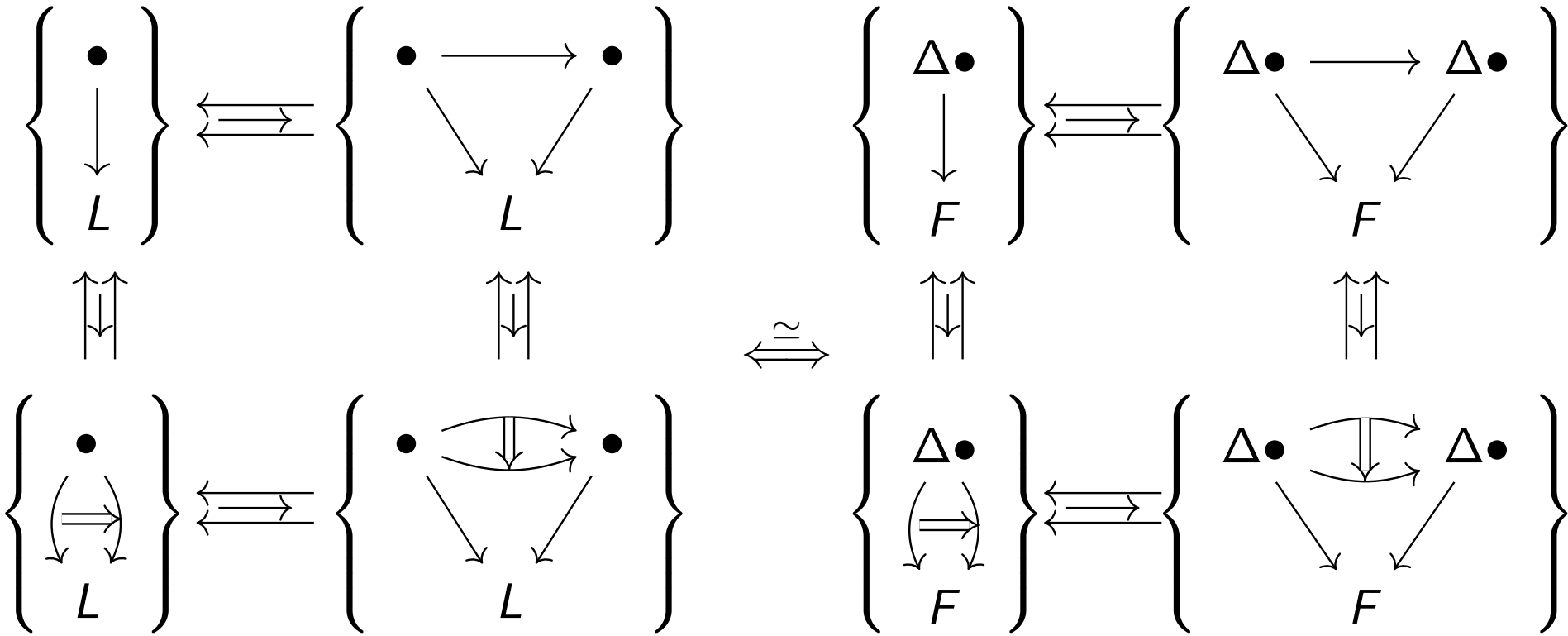
What's the point of this? Well, let $F : I \to \mathcal{C}$ be a 2-functor and L an object ...

L is the Limit of F via the Double Category of Cones



$$\mathcal{C}_{//F}$$

L is the Limit of F via the Double Category of Cones



$\mathcal{C}_{//L}$ $\mathcal{C}_{//F}$

2-Limits via Double Categories

Observation

$\mathcal{C}_{//L}$ has a horizontally terminal object $\mathrm{id}_L : L \to L$.

Theorem (Grandis-Paré)

Let $F : I \to \mathcal{C}$ be a 2-functor. The following are equivalent:

1. *F admits a 2-limit.*
2. *There is an* **equivalence** $\mathcal{C}_{//L} \simeq \mathcal{C}_{//F}$ *for some object L.*
3. *$\mathcal{C}_{//F}$ has a horizontally terminal object.*

What do we mean by equivalence here?

Horizontal Equivalences

The definition of double category of cones does not specify any equivalence, however, our theorem relies on the following:

Definition

A double functor is a **horizontal equivalence** if it induces an equivalence on the underlying first and second underlying horizontal categories.

Proposition

If two double categories are horizontally equivalent then one has a horizontally terminal object if and only if the other has one.

$\Rightarrow$ The theory of 2-limits requires horizontal equivalences.

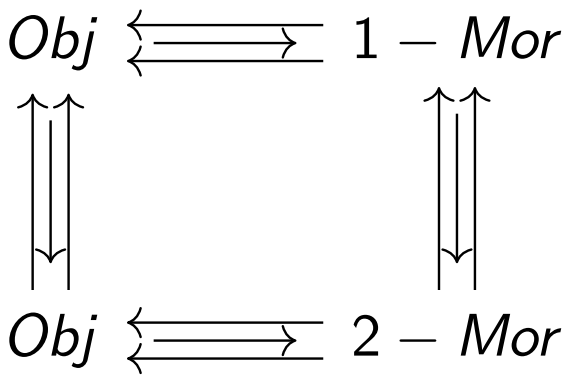
From 2-Categories to Double Categories

We move on to embeddings from 2-categories into double categories!

Theorem

There is a fully faithful functor from 2-categories to double categories. The essential image is given by double categories in which all vertical morphisms are identities.

Here is a rough diagram:



Non-Example: Adjoint Equivalence

$\mathcal{A}\mathrm{dj}\mathcal{E}\mathrm{q}$ denote the 2-category classifying adjoint equivalences.
Recall it is biequivalent to [0].

Remark

The image of the adjoint equivalence is **not** horizontally equivalent to the terminal double category!
Indeed, if it was it would imply the underlying 1-category is equivalent to the terminal object.

So, this embedding will **not** preserve equivalences!

Double Categories up to Horizontal Biequivalence

An alternative is to use a weakening of horizontal equivalences: **horizontal biequivalences**[1].

Definition

A functor of double categories is a *horizontal biequivalence* if it induces a biequivalence on the first and second underlying 2-categories.

Proposition (Moser–Sarazola–Verdugo)

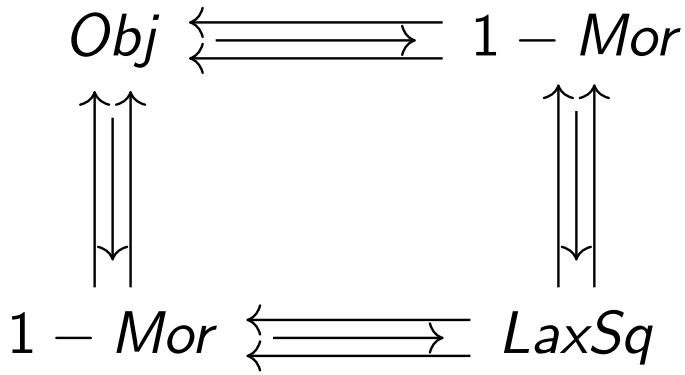
Biequivalent 2-categories induce horizontally biequivalent double categories.

⇒ Double categories arising this way are naturally considered up to horizontal biequivalence.

[1] L. Moser, M. Sarazola and P. Verdugo, A 2Cat-inspired model structure for double categories, Cah. Topol. Géom. Différ. Catég. **63** (2022), no. 2, 184–236

The Square Functor

Moving on to the final example. Given a 2-category $\mathcal{B}$, we can construct a double category $\mathbb{S}\mathrm{q}(\mathcal{B})$, which we can depict as follows:



This construction is naturally symmetric in horizontal and vertical morphisms.

Gregarious Equivalences

We want a notion of equivalence that is **symmetric**!

Definition

A double functor is a *gregarious equivalence* if it is

1. Surjective on objects (up to a certain equivalence relation).
2. Surjective on horizontal and vertical morphisms (up to 2-isomorphisms).
3. Bijective on squares.

Double Categories up to Gregarious Equivalence

This definition is symmetric and we indeed have:[2]

Proposition (Guetta-Moser-Sarazola-Verdugo)

Biequivalent 2-categories induce gregarious equivalent double categories.

$\Rightarrow$ Double categories arising this way are naturally considered up to gregarious equivalence.

[2] L. Guetta, L. Moser, M. Sarazola, P. Verdugo, Fibrantly-induced model structures, arXiv:2301.07801, 2023

Upshot: There are too many Double Categories!

Slogan

The word *double category*, while mathematically precise, can manifest very differently and we are essentially talking about different theories for the same definition.

Question

But wait, isn't that also true for (2-)categories?

We can Compare Equivalences of (2-)Categories ...

For $(2-)$categories, the relevant equivalences are comparable:

Categories:

$$\text{Isomorphisms} < \text{Equivalences}$$

2-**Categories:**

$$\text{Isomorphisms} < \text{Local Equivalence} < \text{Biequivalence}$$

... but Double Categories are not that Nice!

For double categories we have many options that don't compare.

	Hor. Equiv	Hor. Biequiv	Ver. Equiv	Ver. Biequiv	Greg. Equiv
$\mathbb{H}\mathcal{I}\mathrm{so}$	✓	✓	✗	✗	✗
$\mathbb{V}\mathcal{I}\mathrm{so}$	✗	✗	✓	✓	✗
$\mathbb{H}\mathcal{A}\mathrm{dj}\mathcal{E}\mathrm{q}$	✗	✓	✗	✗	✗
$\mathbb{V}\mathcal{A}\mathrm{dj}\mathcal{E}\mathrm{q}$	✗	✗	✗	✓	✗
$\mathcal{S}\mathrm{q}(\mathcal{I}\mathrm{so})$	✓	✓	✓	✓	✓
$\mathcal{S}\mathrm{q}(\mathcal{A}\mathrm{dj}\mathcal{E}\mathrm{q})$	✗	✓	✗	✓	✓

This is just the beginning ...

- We should think of a double category as coming with a specification of "desired equivalence".
- Double categories can be destrictified in at least two ways:
  1. **Strict Double Categories:** Horizontal and vertical composition strictly associative and unital.
  2. **Pseudo Double Categories:** Horizontal composition is strict, but vertical composition is defined up to unitors and associators.
  3. **Doubly Weak Double Categories:** Both compositions are weak.
- We have to consider how our choice of equivalence interacts with the chosen coherence.

Have a Break! Have a KitKat!

Questions?

Comments?

Suggestions?

From Set Theories to Univalent Type Theories

What did we see until now?

- One source of complication is how strict the composition is. This data can be stored in a set theoretical framework.
- Another source of complication is which type of equivalence we desire. Set theoretical framework does not provide us with the tools to address that in our definitions.

$\Rightarrow$ Use different foundations, fitting our aims!

Type Theory: The Part that makes Sense

Slogan (Shulman)

Doing type theory means doing category theory and zooming in so much that one forgets the global perspective on objects and morphisms.

Type theory	**Set theory**	**Category theory**
Types X	Sets	Objects
Terms $x : X$	Elements $x \in X$	$Y \xrightarrow{x} X$
Type constructors	Set constructions	Universal properties

Remark (History)

Goes back to ideas by Russell 1908!

Some Type Constructors

Definition (Unit type)

The unit type 1 has a term $t : 1$ and every other term is equal to it.

Definition (Empty type)

The unit type $\emptyset$ has no terms.

Definition (Product type)

Given two types X, Y, there exists a new type $X \times Y$. A term in $z : X \times Y$ is given as a pair (x, y) with $x : X$, $y : Y$.

Definition (Function type)

Given two types X, Y, there exists a new type $X \to Y$, the *function type*.

Type Theory: The Slightly More Confusing Part

- We saw types as standalone beings.
- We need the ability to types on top of each other, meaning one type is **dependent** on others.
- For that we want a **type of types**.

Universe Types

Definition (Universe types)

A universe type U is a type whose terms $X : U$ represent types in the type theory.

Remark

Such universes are analogous to the *set of all sets* or *category of all categories*.

Remark

This faces the usual problems coming from Russel's paradox, which one has to address in some way.

Dependent Types

Definition (Dependent type)

A dependent type is a function type $X : A \to U$. Here X *depends* on the type A.

Remark

Depending on the context this captures:

- A set-indexed collection of sets.
- A category valued functor.

Dependent Type Constructors: $\sum$ and ...

Dependent types give us access to more powerful type constructors:

Definition ($\sum$-Types)

Given a dependent type $B\colon A \to U$, the sigma type $\sum(a\colon A), B(a)$ is a type with terms given by pairs $(a, x\colon B(a))$.

Example

In the context of sets, if B is an A-indexed collection of sets, then the set $\sum(a\colon A), B(a)$ is simply $\coprod_{a \in A} B(a)$

Example

If B is an A-indexed collection of groupoids, then $\sum(a\colon A), B(a) = \int_A B$, i.e. the *Grothendieck construction*.

... and $\prod$-Types

Definition ($\prod$-Types)

Given a dependent type $B\colon A \to U$, the pi type $\prod(a\colon A), B(a)$ is a type with terms given by functions from A, that take $a\colon A$ to a term in $B(a)$.

Example

In the context of sets, if B is an A-indexed collection of sets, then $\prod(a\colon A), B(a)$ is describing a section of the projection

$$\coprod_{a \in A} B(a) \to A$$

Type Theory: The Really Confusing Part

Remark

Unlike set theory and category theory, there is nothing outside a type theory.

$\Rightarrow$ In particular, this means constructors are also used to prove statements!

Identity Types

Definition

For a given type X and terms $x, y : X$ there is a new **dependent type** (on $X \times X$) denoted $x =_X y$, called the *identity type*.

Definition

Two terms $x, y : X$ are *equal* if the type $x =_X y$ has a term.

Example

Types X, Y are themselves terms in U, hence equal if $X =_U Y$ has a term.

Example

For given type X and term x, we have $\mathrm{refl}_x : x =_X x$. So, every term is equal to itself.

Identity Types in the World

Example

For a given set S, the identity type $S \times S \to \mathcal{S}\mathrm{et}$ is given as

$$(s,t) \mapsto \begin{cases} \{*\} & : s = t \\ \emptyset & : s \neq t \end{cases}$$

and so $\sum_S (s = t)$ is the diagonal map: $\Delta \colon S \to S \times S$.

Example

In a given groupoid $\mathcal{G}$ identities correspond to isomorphisms, hence the identity type

$$(g =_{\mathcal{G}} h) = \mathit{Iso}(g,h) = \mathrm{Hom}_{\mathcal{G}}(g,h).$$

Hence, $\sum_{\mathcal{G}} (g = h) = \mathcal{G}^{[1]}$, the path groupoid.

Quantifiers

Here is an exercise:

Exercise

Let $\varphi(x)$ be a formula over a set S.

1. $\exists(x \in S), \varphi(x)$ *if and only if* $\{x \in S : \varphi(x)\}$ *is non-empty.*
2. $\forall(x \in S), \varphi(x)$ *if and only if* $\{x \in S : \varphi(x)\} \to S$ *has a section.*

$\Rightarrow$ In type theory:

1. Dependent types denote formulas.
2. $\sum$ of a dependent type is also the existential quantifier.
3. $\prod$ of the dependent type is also the universal quantifier.

Set as a model for Type Theories

We can realize these constructors in set theory:

Type theory	**Set theory**	**Logic**
Unit type	Singleton	True
Empty type	Empty set	False
Product type	Product	and
Function type	Set of functions	implies
Universe type	Set of all sets	Set of Truth Values
Dependent type (on A)	Function over A	Formula about A
Identity type $x =_X y$	Diagonal $X \to X \times X$	$=$
$\sum$-Type	Domain	$\exists$
$\prod$-Type	Sections	$\forall$

Reasoning within a Type Theory

Mathematics	Type theory
Magma	$\mathcal{M}\mathrm{ag} = \sum(X : U), X \times X \to X.$
Associative magma	$\mathcal{A}\mathrm{ssoc}\mathcal{M}\mathrm{ag} = \sum(X : U)(m : X \times X \to X),$ $\prod(x\ y\ z : X), m\ (m x\ y) z = m\ x\ (m\ y\ z).$

Lemma

Every associative magma induces a magma.

Proof.

$\pi_1 : \mathit{AssocMag} \to \mathit{Mag}$ □

Types of Identity Types: UIP

Question

How does the identity type behave?

Answer (First guess)

The identity type is either empty or the unit i.e. we have

$$UIP := \prod(X : U)(x\ y : X)(p\ q : x =_X y), (p = q).$$

This does not follow from the axioms and so we can **Uniqueness of Identity Proofs (UIP)**, which states that the type UIP has a term.

Type of Equivalences

Definition (Type of equivalences)

The type of equivalences $A \simeq B$ is defined as

$$\sum (f : A \to B)(g : B \to A)(h : B \to A), (fg = \mathrm{id}) \times (hf = \mathrm{id}).$$

Remark

In sets this corresponds to the set of isomorphisms.
In groupoids this corresponds to the groupoid of equivalences.

Univalence Axiom

Do we have examples of equivalences?

Example

Every identity is an equivalence, so we have a map $A = B \to A \simeq B$.

Definition (Univalence axiom)

A type theory satisfies the *univalence axiom* if the map $A = B \to A \simeq B$ is an equivalence.

$\Rightarrow$ This makes the notion of identity far richer!

Homotopical Aspects of Univalent Foundations

Definition

A type X is (-1)-truncated, or a *proposition*, if $\prod(x\ y : X), x = y$.

Definition

A type X is 0-truncated, or a *set*, if

$$\prod(x\ y : X)(f\ g : x = y), f = g.$$

$\Rightarrow$ More generally we can define arbitrary truncation levels!

Univalent Universes

Univalent universes result in correct identities.

Example

Let U be univalent.

1. Identities in U correspond to equivalences of types.
2. Identities in $\mathcal{M}\mathrm{ag}$ correspond to equivalences of types that respect the operation.
3. Identities in $\mathcal{A}\mathrm{ssoc}\mathcal{M}\mathrm{ag}$ correspond to equivalences of types that respect the operation and associativity.

Remark

Very different from set theory. For example, identities in the set of groups are still just set-theoretic identities!

Univalence in Set Theory

We saw that sets model type theoretical constructions. However, sets are very much not univalent.

- $A \simeq B$: the set of isomorphisms.
- $A = B$: the set of equalities (empty or a point).

$\Rightarrow$ We have univalence if A, B are $\emptyset$ or $\{*\}$.

Univalence in Groupoid Theory

For groupoids we get some aspect of univalence:

- $A \simeq B$: the groupoid of equivalences.
- $A = B$: the set of isomorphisms.

$\Rightarrow$ We have univalence if A, B are discrete (i.e. sets).

Remark

We notice: To get univalence for everything we need to go to ∞-groupoids.

Univalence in ∞-Groupoid Theory

∞-groupoids (i.e. Kan complexes) model univalent type theories.
Most type constructors are interpreted as before.

Some are more interesting:

Kan complex	**Type theory**
Kan fibration	Dependent type
Path fibration	Identity type
n-truncatedness	n-truncatedness

$\Rightarrow A =_U B$ is Equiv(A,B)

Summary

- Type theory is a foundation for mathematics, it can have something to do with computers, but does not have to.
- Type theory internalizes its own language, and particularly the notion of identity.
- Identities in type theories can behave the way we are used to, meaning we have **uniqueness of identity proofs (UIP)**.
- Identities in type theories can be more intricate, for example, by imposing the **univalence axiom**.

Goal

Leverage the stronger identity given by univalence in the study of categories.

Have another Break! Have another KitKat!

Questions?

Comments?

Suggestions?

Defining Categories in Univalent Type Theories

A category is described as follows:

$$
\begin{aligned}
\mathfrak{Cat} := \sum\ & (O : U) \\
& (M : O \to O \to U) \\
& \left(m : \prod(x\ y\ z : O), (M\ x\ y \to M\ y\ z \to M\ x\ z)\right) \\
& \left(e : \prod(x : O), (M\ x\ x)\right), \\
& \quad \left(\mathit{rid} : \prod(x\ y : O)(f : M\ x\ y), m\ f\ (e\ x) = f\right) \\
& \times \left(\mathit{lid} : \prod(x\ y : O)(g : M\ x\ y), m\ (e\ y)\ g = g\right) \\
& \times \left(\mathit{assoc} : \prod(x\ y\ z\ w : O)(f : M\ x\ y)(g : M\ y\ z)(h : M\ z\ w),\right. \\
& \quad \left.(m\ (m\ f\ g)h = m\ f\ (m\ g\ h))\right) \\
& \times \left(\mathit{homsets} : \prod(x\ y : O), (\mathit{isaset}\ M\ x\ y)\right).
\end{aligned}
$$

Identities in $\mathcal{C}\mathrm{at}$

Unwinding the definition of $=$ inside the type $\mathcal{C}\mathrm{at}$, a term in $\mathcal{C} = \mathcal{D}$ corresponds to the following data:

1. A term in $\alpha\colon O_{\mathcal{C}} = O_{\mathcal{D}}$
2. A term in $\beta\colon M_{\mathcal{C}}\ x\ y = M_{\mathcal{D}}\ (\alpha\ x)\ (\alpha\ y)$
3. α and β respect the structure i.e. are "functorial".

How does that correspond to isomorphisms and equivalences in categories?

- (2) always corresponds to being **fully faithful**. ✓
- (1) just implies an equivalence of objects. ✗

Categories up to Isomorphisms: Setcategories

Let $F : \mathcal{C} \to \mathcal{D}$ be a functor.

1. F is an isomorphism if it is fully faithful and a bijection on objects.
2. Equivalences of sets correspond to bijections.

$\Rightarrow$ To get isomorphisms we need a set of objects.

Definition

A *setcategory* is a category with a set of objects

$$\mathcal{C}\mathrm{at}_{set} = \sum (\mathcal{C} : \mathcal{C}\mathrm{at}), (\mathit{isaset}\ O).$$

Proposition (Ahrens–Kapulkin–Lumsdaine)

Identities of setcategories are isomorphisms.

Categories up to Equivalences: Univalent Categories

Let $F : \mathcal{C} \to \mathcal{D}$ be a functor.

- F is an equivalence if it is fully faithful and an equivalence on underlying groupoids.

$\Rightarrow$ Need the objects to correspond to the underlying groupoid.[3]

Definition

A category $\mathcal{C}$ is *univalent* if for all objects c, d, we have that $c =_O d \to c \simeq_{\mathcal{C}} d$ is an equivalence.

$$\mathcal{C}\mathrm{at}_{\mathcal{U}\mathrm{niv}} = \sum(\mathcal{C} : \mathcal{C}\mathrm{at}), (\mathit{isUniv}\ \mathcal{C}).$$

Proposition (Ahrens–Kapulkin–Lumsdaine)

Identities in the type of univalent categories are equivalences.

[3] This heavily borrows from ideas of Rezk.

Defining Double Categories in Type Theories

We can use a combination of $\sum, \prod, \ldots$ to define double categories in type theories, which particularly includes the four types O, Hor, Ver, Sq.

We assume that the squares form a set.

An identity $\mathbb{C} = \mathbb{D}$ corresponds to identities:

1. $\alpha : O_{\mathcal{C}} = O_{\mathcal{D}}$
2. $\beta : Hor_{\mathcal{C}}\ x\ y = Hor_{\mathcal{D}}\ (\alpha\ x)\ (\alpha\ y)$
3. $\gamma : Ver_{\mathcal{C}}\ x\ y = Ver_{\mathcal{D}}\ (\alpha\ x)\ (\alpha\ y)$
4. $\delta : Sq_{\mathcal{C}}\ f\ g\ h\ k = Sq_{\mathcal{D}}\ (\beta\ f)\ (\gamma\ g)\ (\gamma\ h)\ (\beta\ k)$
5. appropriately functorial.

Double Categories up to Iso: Double Setcategories

If we want double categories up to isomorphism, we need everything in sight to be a set.

Definition

A *double setcategory* is a double category with a set of objects, horizontal morphisms, vertical morphisms.

Proposition (Ahrens–North–R.–van der Weide)

Identities in the type of double setcategories correspond to isomorphisms.

Towards Double Categories up to Horizontal Equivalences

Our first guess would be that the first and second underlying horizontal categories need to be univalent. Then (O, Ver) will neither be univalent nor a setcategory!

$\Rightarrow$ We need to relax the composition operation and work with pseudo double categories, meaning only strict composition in the horizontal direction.

Horizontally Univalent Pseudo Double Categories

Definition

A pseudo double category is *horizontally univalent* if the first and second underlying horizontal categories are univalent.

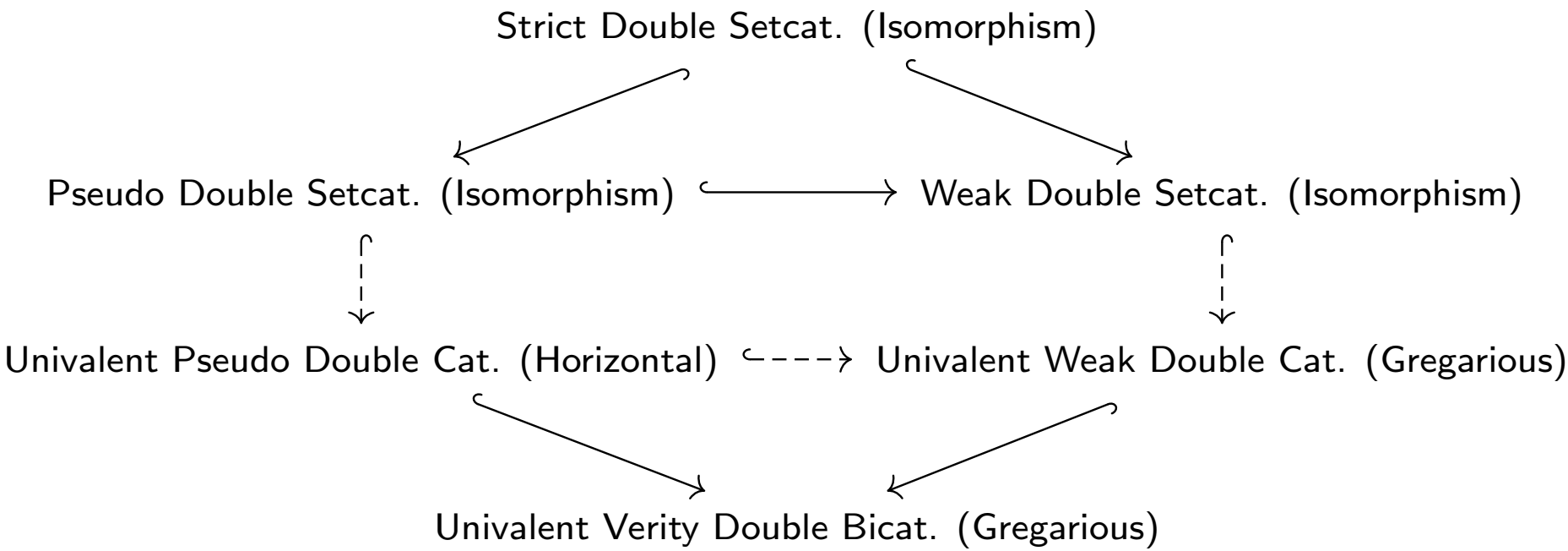
We can use that to define the type of horizontally univalent pseudo double categories.

Proposition (Ahrens–North–R.–van der Weide)

Identities in the type of horizontally univalent pseudo double categories correspond to horizontal equivalences.

Many notions of double categories

We can significantly generalize from here:

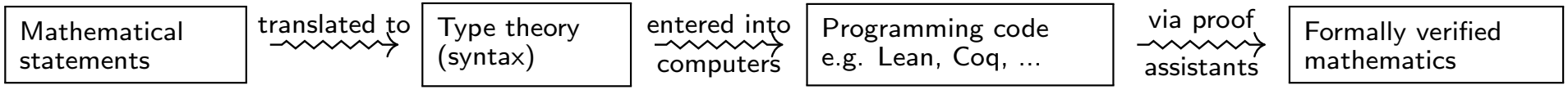


Summary

- Categorical structures come with many equivalences, which cannot be captured by identities in classical mathematics.
- Univalent foundations comes with a **richer notion** of identity.
- Univalent settings permit **refining** the definition of a category.
- Combining the refined definition and the richer identity gives us categorical notions whose identities are precisely the intended equivalences.
- We primarily applied this to (double) categories, but it applies to any other categorical structure (such as 2-categories). We call this the **univalence maxim**.

Some Comments about Formalization I

Formalization aims to formally verify mathematics.



Modern importance of formalization:

- **Liquid Tensor Experiment:** Formalization of condensed mathematics in Lean.[4]
- **Univalence:** Formalization of higher mathematics in univalent foundations motivated by Voevodsky.[5]

[4] https://xenaproject.wordpress.com/2020/12/05/liquid-tensor-experiment/

[5] https://www.ias.edu/ideas/2014/voevodsky-origins

Some Comments about Formalization II

Having identities that fit the structures bring many benefits:

1. We can "transport" results along identities. So, appropriate identities simplify formalizations.
2. In particular, libraries can contain equivalent definitions of the same object and without transport we need to reformalize everything over and over again.
3. For example, the recently growing proof assistant Lean and its library MathLib have UIP. So, there is an imposed "uniqueness of definitions".

The End

Thank you!

- Paper 1: **Univalent Double Categories**, arXiv:2310.09220
  https://arxiv.org/abs/2310.09220
  DOI: https://dl.acm.org/doi/10.1145/3636501.3636955
- Paper 2: **Insights From Univalent Foundations: A Case Study Using Double Categories**, arXiv:2402.05265
  https://arxiv.org/abs/2402.05265
- Formalization:
  https://github.com/UniMath/UniMath/tree/master/UniMath/Bicategories/DoubleCategories
- Questions:
  1. Ask me in person!
  2. Email: nima.rasekh@uni-greifswald.de