

Higher structures via homotopy type theory

Paige Randall North

8-9 September 2026

Outline

- 1 The equivalence principle
- 2 Dependent type theory
- 3 Univalent foundations
- 4 Indiscernibility by example
- 5 The univalence principle¹
- 6 Application: double categories²

¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

The equivalence principle

Equivalence principle

Reasoning in mathematics should be **invariant under** the appropriate notion of **equivalence**.

The equivalence principle

Equivalence principle

Reasoning in mathematics should be **invariant under** the appropriate notion of **equivalence**.

Notion of equivalence depends on the objects under consideration:

- **equal** numbers, functions, ...
- **isomorphic** sets, groups, rings, ...
- **equivalent** categories
- **biequivalent** bicategories
- ...

Non-examples: statements violating the equivalence principle

We can easily **violate** this principle:

Exercise

Find a statement about sets that is not invariant under isomorphism:

$$\{\emptyset, \{\emptyset\}\} \cong \{\emptyset, \{\{\emptyset\}\}\}$$

Exercise

Find a statement about categories that is not invariant under equivalence:



Non-examples: statements violating the equivalence principle

We can easily **violate** this principle:

Exercise

Find a statement about sets that is not invariant under isomorphism:

$$\{\emptyset, \{\emptyset\}\} \cong \{\emptyset, \{\{\emptyset\}\}\}$$

$$\{\emptyset\} \in X$$

Exercise

Find a statement about categories that is not invariant under equivalence:



$\mathcal{C}$ has exactly 1 object.

A language for invariant properties

Michael Makkai, *Towards a Categorical Foundation of Mathematics*:

*“The basic character of the Principle of Isomorphism is that of a **constraint on the language** of Abstract Mathematics; a welcome one, since it provides for the separation of sense from non-sense.”*

A language for invariant properties

Michael Makkai, *Towards a Categorical Foundation of Mathematics*:

*“The basic character of the Principle of Isomorphism is that of a **constraint on the language** of Abstract Mathematics; a welcome one, since it provides for the separation of sense from non-sense.”*

Goal

To have a **syntactic criterion** for properties and constructions that are invariant under equivalence

How to break the equivalence principle for categories. . .

- Recall: the statement

The category $\mathcal{C}$ has exactly one object.

is not invariant under equivalence of categories.

- In general, referring to **equality of objects** breaks invariance, but. . .

How to break the equivalence principle for categories. . .

- Recall: the statement

The category $\mathcal{C}$ has exactly one object.

is not invariant under equivalence of categories.

- In general, referring to **equality of objects** breaks invariance, but. . .
- even the **definition** of category refers to equality of objects:

Problem

“If $\text{dom}(g)$ is **equal to** $\text{cod}(f)$, then $g \circ f$ exists.”

How to break the equivalence principle for categories. . .

- Recall: the statement

The category $\mathcal{C}$ has exactly one object.

is not invariant under equivalence of categories.

- In general, referring to **equality of objects** breaks invariance, but. . .
- even the **definition** of category refers to equality of objects:

Problem

“If $\text{dom}(g)$ is **equal to** $\text{cod}(f)$, then $g \circ f$ exists.”

Can we give a definition of category without using equality of objects?

... and how to fix it.

Solution

Use a logic/language of **dependent sets**, in which $\text{dom}(g) = \text{cod}(f)$ is encoded by what type of thing f and g are.

... and how to fix it.

Solution

Use a logic/language of **dependent sets**, in which $\text{dom}(g) = \text{cod}(f)$ is encoded by what type of thing f and g are.

A category consists of

- a set O of objects
- for each $x, y \in O$, a type/set $A(x, y)$ of arrows
- for each $x, y, z \in O$ and each $f \in A(x, y)$ and $g \in A(y, z)$, a composite $g \circ f \in A(x, z)$
- for each $x \in O$, an identity $\text{id}_x \in A(x, x)$
- ...

... and how to fix it.

Solution

Use a logic/language of **dependent sets**, in which $\text{dom}(g) = \text{cod}(f)$ is encoded by what type of thing f and g are.

A category consists of

- a set O of objects
- for each $x, y \in O$, a type/set $A(x, y)$ of arrows
- for each $x, y, z \in O$ and each $f \in A(x, y)$ and $g \in A(y, z)$, a composite $g \circ f \in A(x, z)$
- for each $x \in O$, an identity $\text{id}_x \in A(x, x)$
- ...

Gives rise to a **dependently typed language** by adding logical connectives.

Invariance for statements

Theorem (Freyd '76, Blanc '78)

A property of categories (expressed in 2-sorted first-order logic) is invariant under equivalence iff it can be expressed in this dependently typed language, using equality for arrows but not for objects.

Invariance for statements

Theorem (Freyd '76, Blanc '78)

*A **property** of categories (expressed in 2-sorted first-order logic) is invariant under equivalence iff it can be expressed in this dependently typed language, using equality for arrows but not for objects.*

- What about **constructions** on categories?

Invariance for statements

Theorem (Freyd '76, Blanc '78)

*A **property** of categories (expressed in 2-sorted first-order logic) is invariant under equivalence iff it can be expressed in this dependently typed language, using equality for arrows but not for objects.*

- What about **constructions** on categories?
- What about other mathematical structures?

Outline

- 1 The equivalence principle
- 2 Dependent type theory**
- 3 Univalent foundations
- 4 Indiscernibility by example
- 5 The univalence principle¹
- 6 Application: double categories²

¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a self-sufficient foundation of mathematics
 - Well-suited for machine verification

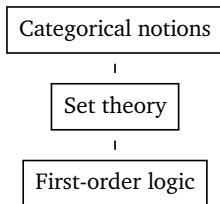
Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a **self-sufficient** foundation of mathematics
 - Well-suited for machine verification

Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a **self-sufficient** foundation of mathematics
 - Well-suited for machine verification

Classical mathematics:

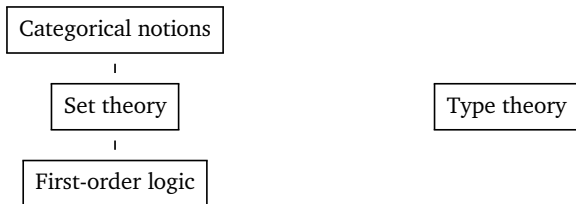


Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a **self-sufficient** foundation of mathematics
 - Well-suited for machine verification

Classical mathematics:

Mathematics à la Martin-Löf:

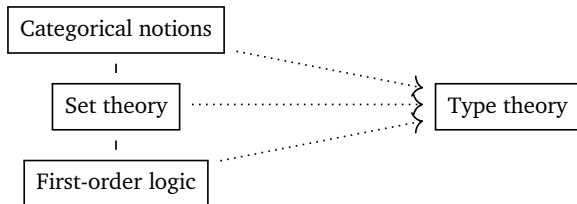


Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a **self-sufficient** foundation of mathematics
 - Well-suited for machine verification

Classical mathematics:

Mathematics à la Martin-Löf:

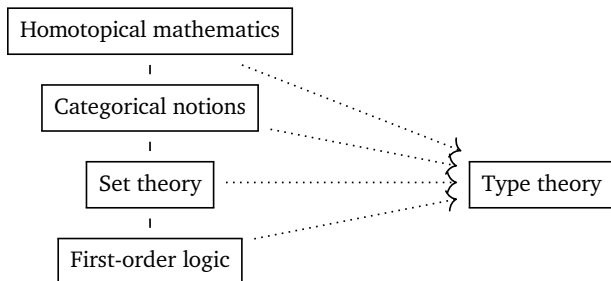


Type theory's beginnings

- 1970s: Martin-Löf introduces his type theory
 - As a **self-sufficient** foundation of mathematics
 - Well-suited for machine verification

Classical mathematics:

Mathematics à la Martin-Löf:



What does type theory look like?

In mathematics, statements look like the following:

- Consider propositions P, Q . P follows from $P \wedge Q$.
- Consider a natural number n . The sum $n + n$ is even.
- Consider a category C , a presheaf P on C , and an object X of C .
The set of natural transformations from the Yoneda embedding of X to P is isomorphic to the set PX .
- Consider a space X . The cone on X is contractible.

In type theory, we write this as

- $c : P \times Q \vdash p : P$
- $n : \mathbb{N} \vdash e(n) : \text{isEven}(n + n)$
- $C : \text{Cat}, P : \widehat{C}, X : \text{ob}(C) \vdash i : \text{hom}_{\widehat{C}}(y_X, P) \cong PX$
- $X : \text{Spaces} \vdash c(X) : \text{isContr}(CX)$

Constituents are types, terms, and contexts.

What does type theory look like?

In mathematics, statements look like the following:

- Consider propositions P, Q . P follows from $P \wedge Q$.
- Consider a natural number n . The sum $n + n$ is even.
- Consider a category C , a presheaf P on C , and an object X of C .
The set of natural transformations from the Yoneda embedding of X to P is isomorphic to the set PX .
- Consider a space X . The cone on X is contractible.

In type theory, we write this as

- $c : P \times Q \vdash p : P$
- $n : \mathbb{N} \vdash e(n) : \text{isEven}(n + n)$
- $C : \text{Cat}, P : \widehat{C}, X : \text{ob}(C) \vdash i : \text{hom}_{\widehat{C}}(y_X, P) \cong PX$
- $X : \text{Spaces} \vdash c(X) : \text{isContr}(CX)$

Constituents are types, terms, and contexts.

What does type theory look like?

In mathematics, statements look like the following:

- Consider propositions P, Q . P follows from $P \wedge Q$.
- Consider a natural number n . The sum $n + n$ is even.
- Consider a category C , a presheaf P on C , and an object X of C .
The set of natural transformations from the Yoneda embedding of X to P is isomorphic to the set PX .
- Consider a space X . The cone on X is contractible.

In type theory, we write this as

- $c : P \times Q \vdash p : P$
- $n : \mathbb{N} \vdash e(n) : \text{isEven}(n + n)$
- $C : \text{Cat}, P : \widehat{C}, X : \text{ob}(C) \vdash i : \text{hom}_{\widehat{C}}(y_X, P) \cong PX$
- $X : \text{Spaces} \vdash c(X) : \text{isContr}(CX)$

Constituents are types, terms, and contexts.

What does type theory look like?

In mathematics, statements look like the following:

- Consider propositions P, Q . P follows from $P \wedge Q$.
- Consider a natural number n . The sum $n + n$ is even.
- Consider a category C , a presheaf P on C , and an object X of C .
The set of natural transformations from the Yoneda embedding of X to P is isomorphic to the set PX .
- Consider a space X . The cone on X is contractible.

In type theory, we write this as

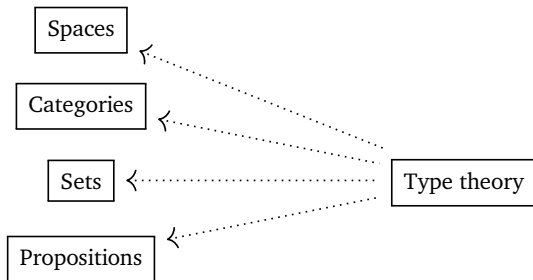
- $c : P \times Q \vdash p : P$
- $n : \mathbb{N} \vdash e(n) : \text{isEven}(n + n)$
- $C : \text{Cat}, P : \widehat{C}, X : \text{ob}(C) \vdash i : \text{hom}_{\widehat{C}}(y_X, P) \cong PX$
- $X : \text{Spaces} \vdash c(X) : \text{isContr}(CX)$

Constituents are types, terms, and contexts.

Interpretations of type theory into classical mathematics

Classical mathematics:

Mathematics à la Martin-Löf:



Interpretations of type theory into classical mathematics

Types	Terms	Product	Equality
Propositions	proofs	$\wedge$	$=$
Sets	elements	$\times$	$=$
Categories	objects	$\times$	$\cong$
Spaces	points	$\times$	$\sim$

Interpretations of type theory into classical mathematics

Types	Terms	Product	Equality
Propositions	proofs	$\wedge$	$=$
Sets	elements	$\times$	$=$
Categories	objects	$\times$	$\cong$
Spaces	points	$\times$	$\sim$

- Everything in type theory respects equality
 - This is the definition of equality in type theory (roughly)

Interpretations of type theory into classical mathematics

Types	Terms	Product	Equality
Propositions	proofs	$\wedge$	$=$
Sets	elements	$\times$	$=$
Categories	objects	$\times$	$\cong$
Spaces	points	$\times$	$\sim$

- Everything in type theory respects equality
 - This is the definition of equality in type theory (roughly)
- Mathematics in type theory
 - + Interpretation into X s where equality is interpreted by Y
 - $\rightsquigarrow$ Mathematics in X s up to Y

Interpretations of type theory into classical mathematics

Types	Terms	Product	Equality
Propositions	proofs	$\wedge$	$=$
Sets	elements	$\times$	$=$
Groupoids	objects	$\times$	$\approx$
Spaces	points	$\times$	$\sim$

- Everything in type theory respects equality
 - This is the definition of equality in type theory (roughly)
- Mathematics in type theory
 - + Interpretation into X s where equality is interpreted by Y
 - $\rightsquigarrow$ Mathematics in X s up to Y

Different notions of equality

Synthetic vs. analytic equalities

In type theory with the equality type, we always have a (“synthetic”) equality type between $a, b : D$

$$a =_D b.$$

Depending on the type D , we might also have a type of “analytic” equalities

$$a \simeq_D b.$$

A *univalence principle* for this D and this $\simeq_D$ states that

$$(a =_D b) \rightarrow (a \simeq_D b)$$

is an equivalence.

Identicals and indiscernibles

Identity of indiscernibles

Leibniz: two things are equal when they are *indiscernible* (have the same properties).

$$(a = b) \leftarrow (\forall P. P(a) \leftrightarrow P(b))$$

Identicals and indiscernibles

Identity of indiscernibles

Leibniz: two things are equal when they are *indiscernible* (have the same properties).

$$(a = b) \leftrightarrow (\forall P. P(a) \leftrightarrow P(b))$$

Identicals and indiscernibles

Identity of indiscernibles

Leibniz: two things are equal when they are *indiscernible* (have the same properties).

$$(a =_D b) \leftrightarrow \left(\prod_{P:D \rightarrow \text{Type}} P(a) \simeq P(b) \right)$$

Identicals and indiscernibles

Identity of indiscernibles

Leibniz: two things are equal when they are *indiscernible* (have the same properties).

$$(a =_D b) \leftrightarrow \left(\prod_{P:D \rightarrow \mathsf{Type}} P(a) \simeq P(b) \right)$$

- This holds in type theory.

Identicals and indiscernibles

Identity of indiscernibles

Leibniz: two things are equal when they are *indiscernible* (have the same properties).

$$(a =_D b) \leftrightarrow \left(\prod_{P:D \rightarrow \mathsf{Type}} P(a) \simeq P(b) \right)$$

- This holds in type theory.
- Given a univalence principle $(a =_D b) \simeq (a \simeq_D b)$, we find an *equivalence principle*:

$$(a \simeq_D b) \rightarrow \left(\prod_{P:D \rightarrow \mathsf{Type}} P(a) \simeq P(b) \right)$$

so everything respects $\simeq_D$.

Goal state

Our goal now is to find *univalence principles*

$$(a =_D b) \simeq (a \simeq_D b)$$

in our type theory: e.g.,

- $(C =_{\text{Set}} D) \simeq (C \cong D)$
- $(C =_{\text{Grp}} D) \simeq (C \cong D)$
- $(C =_{\text{Cat}} D) \simeq (C \simeq D)$
- $(C =_{\text{biCat}} D) \simeq (C \simeq D)$

Then all reasoning in the type theory satisfies the equivalence principle, i.e., is not evil.

Outline

- 1 The equivalence principle
- 2 Dependent type theory
- 3 Univalent foundations**
- 4 Indiscernibility by example
- 5 The univalence principle¹
- 6 Application: double categories²

¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

Univalence

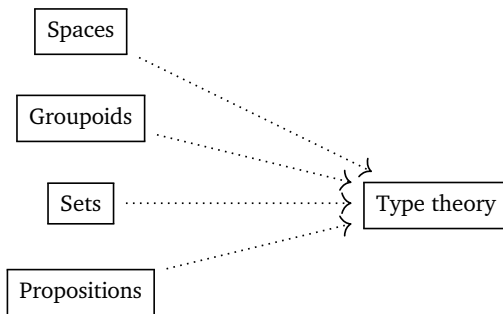
- We've seen that equality in type theory can be interpreted as notions weaker than classical equality (e.g. isomorphism, paths).
- Voevodsky imported weakness for equality from the interpretation in spaces into type theory by imposing the *Univalence Axiom* (UA):

The canonical function $(A =_{\text{Type}} B) \rightarrow (A \simeq B)$ is an equivalence of types, for any types A and B .

- UA is validated by the interpretation into spaces, but not into propositions, sets, or groupoids.
- Instead we **internalize** these notions.

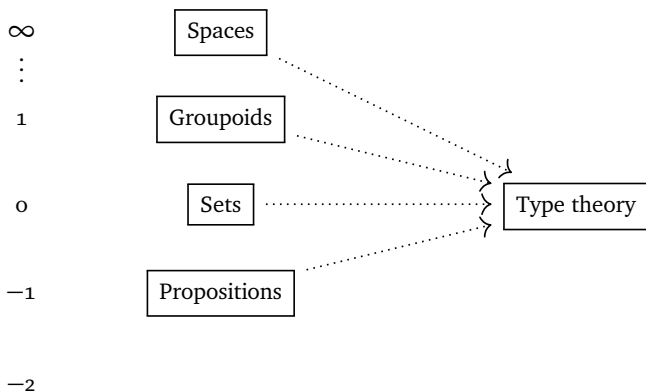
Internalization of classical mathematics into type theory

Classical mathematics: **Mathematics à la Martin-Löf:**



Internalization of classical mathematics into type theory

Classical mathematics: **Mathematics à la Martin-Löf:**



Roughly: A type T has h-level n if $\pi_m(T) = *$ for all $m > n$.

More precisely: $\text{ishlevel } (-2) T := \Sigma_{t:T} \Pi_{x:T} x = t$

$\text{ishlevel } (n + 1) T := \Pi_{x,y:T} \text{ishlevel } n (x = y).$

Univalent mathematics

- If two types A, B are propositions,

$$(A \simeq B) \simeq (A \leftrightarrow B)$$

Univalent mathematics

- If two types A, B are propositions,

$$(A =_{\text{Prop}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \leftrightarrow B)$$

so everything respects bi-implication of propositions.

Univalent mathematics

- If two types A, B are propositions,

$$(A =_{\text{Prop}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \leftrightarrow B)$$

so everything respects bi-implication of propositions.

- If A, B are sets,

$$(A =_{\text{Set}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \cong B)$$

so everything respects bijection of sets.

Univalent mathematics

- If two types A, B are propositions,

$$(A =_{\text{Prop}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \leftrightarrow B)$$

so everything respects bi-implication of propositions.

- If A, B are sets,

$$(A =_{\text{Set}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \cong B)$$

so everything respects bijection of sets.

- For types A, B which are structured sets (groups, rings, etc.),

$$(A =_{\text{Grp}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \cong B)$$

so everything respects isomorphism of groups (or rings, etc.).¹

¹Coquand-Danielsson 2013

Univalent mathematics

- If two types A, B are propositions,

$$(A =_{\text{Prop}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \leftrightarrow B)$$

so everything respects bi-implication of propositions.

- If A, B are sets,

$$(A =_{\text{Set}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \cong B)$$

so everything respects bijection of sets.

- For types A, B which are structured sets (groups, rings, etc.),

$$(A =_{\text{Grp}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \cong B)$$

so everything respects isomorphism of groups (or rings, etc.).¹

- For *univalent* categories A, B ,

$$(A =_{\text{UCat}} B) \stackrel{\text{UA}}{\simeq} (A \simeq B) \simeq (A \simeq B)$$

so everything respects equivalence of univalent categories.²

¹Coquand-Danielsson 2013

²Ahrens-Kapulkin-Shulman 2015

Univalent mathematics

- Voevodsky dreamt of ‘univalent mathematics’ in which

$$(A =_D B) \simeq (A \simeq_D B)$$

where D is any type of mathematical object (propositions, sets, groups, categories, ∞ -categories, etc.) and $\simeq_D$ is the appropriate notion of ‘sameness’ for that type of object.

- This would give us an appropriate language in which to study D .

Outline

- 1 The equivalence principle
- 2 Dependent type theory
- 3 Univalent foundations
- 4 Indiscernibility by example**
- 5 The univalence principle¹
- 6 Application: double categories²

¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

Categories

Theorem (univalence for categories; Ahrens-Kapulkin-Shulman)

Given two *univalent* categories C and D ,

$$(C =_{\mathbf{UCat}} D) \simeq (C \simeq D).$$

Definition

A category C is *univalent* if $(x =_{\mathbf{ob}(C)} y) \simeq (x \cong y)$ for all objects x, y .

Categories

Theorem (univalence for categories; Ahrens-Kapulkin-Shulman)

Given two *univalent* categories C and D ,

$$(C =_{\mathbf{UCat}} D) \simeq (C \simeq D).$$

Definition

A category C is *univalent* if $(x =_{\mathbf{ob}(C)} y) \simeq (x \cong y)$ for all objects x, y .

Indiscernibility

We want to generalize the notion of isomorphism to other structures, and we will call this *indiscernibility*.

Preorders

Preorders

A *preorder* is a set P and a reflexive, transitive relation $\leq: P \times P \rightarrow \text{Prop}$. Two elements p, q of a preorder P are *indiscernible* if

$$\prod_{x:P} (p \leq x \leftrightarrow q \leq x) \times (x \leq p \leftrightarrow x \leq q) \times (p \leq p \leftrightarrow q \leq q)$$

or, equivalently, if $(p \leq q) \times (q \leq p)$.

We get two notions of equivalence of preorders:

1. $P \cong_e Q$ (isomorphism up to equality)
2. $P \cong_i Q$

Magmas

Magmas

A *magma* is a set M and a binary operation $M \times M \rightarrow M$. Two elements m, n are *indiscernible* if

$$\prod_{x,y:M} (mx = nx) \times (xm = xn) \times ((xy = m) \leftrightarrow (xy = n)).$$

We get two notions of equivalence of magmas:

1. $M \cong_e N$ (isomorphism up to equality)
2. $M \cong_i N$ (isomorphism up to indiscernibility)

A lower structure identity principle in UF

Theorem (univalence for magmas with $\cong_e$)

Given two magmas M, N ,

$$(M =_{\text{Mag}} N) \simeq (M \cong_e N).$$

- This is a special case of univalence for sets with structure (Coquand-Danielsson)
- The same holds for preorders with $\cong_e$.

Another lower structure identity principle in UF?

Univalence with $\cong_i$

Q: Can we hope for the same with $\cong_i$?

Another lower structure identity principle in UF?

Univalence with $\cong_i$

Q: Can we hope for the same with $\cong_i$?

A: No: for example, the projection $U : \mathbf{Proset} \rightarrow \mathbf{Set}$ would then take an equivalence $P \cong_i Q$ to an equality/equivalence $P \cong_e Q$.

For example, let **1** be the preorder whose underlying set has one element, and let **2** be the preorder whose underlying set has two elements a and b for which $a \leq b$ and $b \leq a$.



Another lower structure identity principle in UF?

Univalence with $\cong_i$

Q: Can we hope for the same with $\cong_i$?

A: No: for example, the projection $U : \mathbf{Proset} \rightarrow \mathbf{Set}$ would then take an equivalence $P \cong_i Q$ to an equality/equivalence $P \cong_e Q$.

A: Yes: if we identify equality and indiscernibility.

For example, let **1** be the preorder whose underlying set has one element, and let **2** be the preorder whose underlying set has two elements a and b for which $a \leq b$ and $b \leq a$.



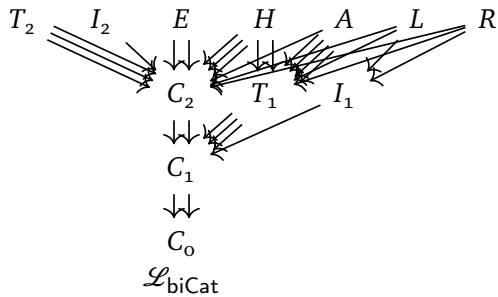
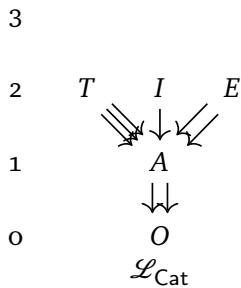
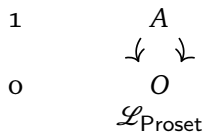
Outline

- 1 The equivalence principle
- 2 Dependent type theory
- 3 Univalent foundations
- 4 Indiscernibility by example
- 5 The univalence principle¹**
- 6 Application: double categories²

¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

Signatures



Structures

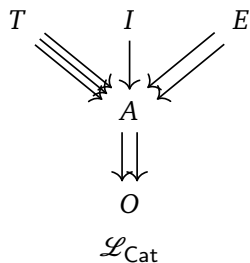
- Morally, an $\mathcal{L}$ -**structure** for a signature $\mathcal{L}$ is a Reedy fibrant diagram $\mathcal{L} \rightarrow \mathbf{Type}$.

Structures

- Morally, an $\mathcal{L}$ -**structure** for a signature $\mathcal{L}$ is a Reedy fibrant diagram $\mathcal{L} \rightarrow \text{Type}$.
- In type theory, we define an $\mathcal{L}$ -**structure** fiberwise.

Structures

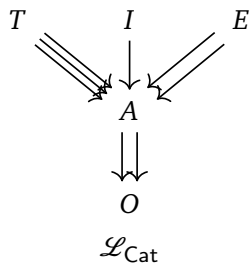
- Morally, an $\mathcal{L}$ -**structure** for a signature $\mathcal{L}$ is a Reedy fibrant diagram $\mathcal{L} \rightarrow \text{Type}$.
- In type theory, we define an $\mathcal{L}$ -**structure** fiberwise.
- An $\mathcal{L}_{\text{Cat}}$ -structure $\mathcal{C}$ consists of:



- $\mathcal{C}O : \text{Type}$
- $x, y : \mathcal{C}O \vdash \mathcal{C}A(x, y) : \text{Type}$
- $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash \mathcal{C}I_x(f) : \text{Type}$
- $x, y, z : \mathcal{C}O, f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z) \vdash \mathcal{C}T_{x,y,z}(f, g, h) : \text{Type}$
- $x, y : \mathcal{C}O, f, g : \mathcal{C}A(x, y) \vdash \mathcal{C}E_{x,y}(f, g) : \text{Type}$

Structures

- Morally, an $\mathcal{L}$ -**structure** for a signature $\mathcal{L}$ is a Reedy fibrant diagram $\mathcal{L} \rightarrow \text{Type}$.
- In type theory, we define an $\mathcal{L}$ -**structure** fiberwise.
- An $\mathcal{L}_{\text{Cat}}$ -structure $\mathcal{C}$ consists of:



- $\mathcal{C}O : \text{Type}$
 - $x, y : \mathcal{C}O \vdash \mathcal{C}A(x, y) : \text{Type}$
 - $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash \mathcal{C}I_x(f) : \text{Type}$
 - $x, y, z : \mathcal{C}O, f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z) \vdash \mathcal{C}T_{x,y,z}(f, g, h) : \text{Type}$
 - $x, y : \mathcal{C}O, f, g : \mathcal{C}A(x, y) \vdash \mathcal{C}E_{x,y}(f, g) : \text{Type}$
- Then we add axioms.

Levelwise equivalence

Proposition

For two $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T)$$

where $\cong_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence.

Levelwise equivalence

Proposition

For two $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T)$$

where $\cong_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence.

A levelwise equivalence $\mathcal{C} \cong_{\mathcal{L}\text{Cat-Str}} \mathcal{D}$ consists of:

- $e_O : \mathcal{C}O \xrightarrow{\sim} \mathcal{D}O$
- $x, y : \mathcal{C}O \vdash e_A : \mathcal{C}A(x, y) \xrightarrow{\sim} \mathcal{D}A(e_O x, e_O y)$
- $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash e_I : \mathcal{C}I_x(f) \xrightarrow{\sim} \mathcal{D}I_{e_O x}(e_A f)$
- $x, y, z : \mathcal{C}O, f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z) \vdash e_T : \mathcal{C}T_{x,y,z}(f, g, h) \xrightarrow{\sim} \mathcal{D}T_{e_O x, e_O y, e_O z}(e_A f, e_A g, e_A h)$
- $x, y : \mathcal{C}O, f, g : \mathcal{C}A(x, y) \vdash e_E : \mathcal{C}E_{x,y}(f, g) \xrightarrow{\sim} \mathcal{D}E_{e_O x, e_O y}(e_A f, e_A g)$

Levelwise equivalence

Proposition

For two $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T)$$

where $\cong_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence.

A levelwise equivalence $\mathcal{C} \cong_{\mathcal{L}\text{Cat-Str}} \mathcal{D}$ consists of:

- $e_O : \mathcal{C}O \xrightarrow{\sim} \mathcal{D}O$
- $x, y : \mathcal{C}O \vdash e_A : \mathcal{C}A(x, y) \xrightarrow{\sim} \mathcal{D}A(e_Ox, e_Oy)$
- $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash e_I : \mathcal{C}I_x(f) \xrightarrow{\sim} \mathcal{D}I_{e_Ox}(e_Af)$
- $x, y, z : \mathcal{C}O, f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z) \vdash e_T : \mathcal{C}T_{x,y,z}(f, g, h) \xrightarrow{\sim} \mathcal{D}T_{e_Ox, e_Oy, e_Oz}(e_Af, e_Ag, e_Ah)$
- $x, y : \mathcal{C}O, f, g : \mathcal{C}A(x, y) \vdash e_E : \mathcal{C}E_{x,y}(f, g) \xrightarrow{\sim} \mathcal{D}E_{e_Ox, e_Oy}(e_Af, e_Ag)$

But this is not an equivalence of categories.

Levelwise equivalence

Proposition

For two $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T)$$

where $\cong_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence.

A levelwise equivalence $\mathcal{C} \cong_{\mathcal{L}\text{Cat-Str}} \mathcal{D}$ consists of:

- $e_O : \mathcal{C}O \xrightarrow{\sim} \mathcal{D}O$
- $x, y : \mathcal{C}O \vdash e_A : \mathcal{C}A(x, y) \xrightarrow{\sim} \mathcal{D}A(e_O x, e_O y)$
- $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash e_I : \mathcal{C}I_x(f) \xrightarrow{\sim} \mathcal{D}I_{e_O x}(e_A f)$
- $x, y, z : \mathcal{C}O, f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z) \vdash e_T : \mathcal{C}T_{x,y,z}(f, g, h) \xrightarrow{\sim} \mathcal{D}T_{e_O x, e_O y, e_O z}(e_A f, e_A g, e_A h)$
- $x, y : \mathcal{C}O, f, g : \mathcal{C}A(x, y) \vdash e_E : \mathcal{C}E_{x,y}(f, g) \xrightarrow{\sim} \mathcal{D}E_{e_O x, e_O y}(e_A f, e_A g)$

But this is not an equivalence of categories.

And is it appropriate to call $\mathcal{C}, \mathcal{D}$ categories?

Indiscernibility

Definition

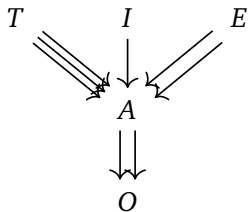
Given an $\mathcal{L}$ -structure M , and an object S of $\mathcal{L}$, we say that two elements $x, y : MS$ are *indiscernible* if substituting x for y in any object of $\mathcal{L}$ that depends on (i.e., has a morphism to) S produces equivalent types.

Definition

An $\mathcal{L}$ -structure M is *univalent* if for any object S of $\mathcal{L}$, and any $x, y : MS$, the type of indiscernibilities between x and y is equivalent to the type of equalities between x and y .

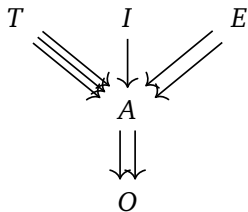
Univalent $\mathcal{L}_{\text{Cat}}$ -structures

Let $\mathcal{C}$ be a univalent $\mathcal{L}_{\text{Cat}}$ -structure.



Univalent $\mathcal{L}_{\text{Cat}}$ -structures

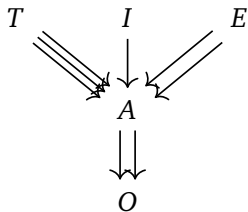
Let $\mathcal{C}$ be a univalent $\mathcal{L}_{\text{Cat}}$ -structure.



- Any two terms $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash i, j : \mathcal{C}I_x(f)$ are indiscernible.
- Each $\mathcal{C}I_x(f)$ is a proposition.
- Similarly, each $\mathcal{C}T_{x,y,z}(f, g, h)$, $\mathcal{C}E_{x,y}(f, g)$ is a proposition.

Univalent $\mathcal{L}_{\text{Cat}}$ -structures

Let $\mathcal{C}$ be a univalent $\mathcal{L}_{\text{Cat}}$ -structure.



- Any two terms $x : \mathcal{C}O, f : \mathcal{C}A(x, x) \vdash i, j : \mathcal{C}I_x(f)$ are indiscernible.
 - Each $\mathcal{C}I_x(f)$ is a proposition.
 - Similarly, each $\mathcal{C}T_{x,y,z}(f, g, h)$, $\mathcal{C}E_{x,y}(f, g)$ is a proposition.
- In the axioms for a category, we have that E behaves like equality (is reflexive and a congruence for T, I, E).
 - Univalence at A means that $f = g$ is equivalent to $\mathcal{C}E_{x,y}(f, g)$.
 - $\mathcal{C}A(x, y)$ is a set.

Univalent $\mathcal{L}_{\text{Cat}}$ -structures

- The indiscernibilities between $a, b : \mathcal{C}O$ consist of
 - $\phi_{x\bullet} : \mathcal{C}A(x, a) \cong \mathcal{C}A(x, b)$ for each $x : \mathcal{C}O$
 - $\phi_{\bullet z} : \mathcal{C}A(a, z) \cong \mathcal{C}A(b, z)$ for each $z : \mathcal{C}O$
 - $\phi_{\bullet\bullet} : \mathcal{C}A(a, a) \cong \mathcal{C}A(b, b)$
 - The following for all appropriate w, x, y, z, f, g, h :

$$\begin{array}{ll}
 \mathcal{C}T_{x,y,a}(f, g, h) \leftrightarrow \mathcal{C}T_{x,y,b}(f, \phi_{y\bullet}(g), \phi_{x\bullet}(h)) & \mathcal{C}I_a(f) \leftrightarrow \mathcal{C}I_b(\phi_{\bullet\bullet}(f)) \\
 \mathcal{C}T_{x,a,z}(f, g, h) \leftrightarrow \mathcal{C}T_{x,b,z}(\phi_{x\bullet}(f), \phi_{\bullet z}(g), h) & \mathcal{C}E_{x,a}(f, g) \leftrightarrow \mathcal{C}E_{x,b}(\phi_{x\bullet}(f), \phi_{x\bullet}(g)) \\
 \mathcal{C}T_{a,z,w}(f, g, h) \leftrightarrow \mathcal{C}T_{b,z,w}(\phi_{\bullet z}(f), g, \phi_{\bullet w}(h)) & \mathcal{C}E_{a,x}(f, g) \leftrightarrow \mathcal{C}E_{b,x}(\phi_{\bullet x}(f), \phi_{\bullet x}(g)) \\
 \mathcal{C}T_{x,a,a}(f, g, h) \leftrightarrow \mathcal{C}T_{x,b,b}(\phi_{x\bullet}(f), \phi_{\bullet\bullet}(g), \phi_{x\bullet}(h)) & \mathcal{C}E_{a,a}(f, g) \leftrightarrow \mathcal{C}E_{b,b}(\phi_{\bullet\bullet}(f), \phi_{\bullet\bullet}(g)) \\
 \mathcal{C}T_{a,x,a}(f, g, h) \leftrightarrow \mathcal{C}T_{b,x,b}(\phi_{\bullet x}(f), \phi_{x\bullet}(g), \phi_{\bullet\bullet}(h)) & \\
 \mathcal{C}T_{a,a,x}(f, g, h) \leftrightarrow \mathcal{C}T_{b,b,x}(\phi_{\bullet\bullet}(f), \phi_{\bullet x}(g), \phi_{\bullet x}(h)) & \\
 \mathcal{C}T_{a,a,a}(f, g, h) \leftrightarrow \mathcal{C}T_{b,b,b}(\phi_{\bullet\bullet}(f), \phi_{\bullet\bullet}(g), \phi_{\bullet\bullet}(h)) &
 \end{array}$$

Univalent $\mathcal{L}_{\text{Cat}}$ -structures

- The indiscernibilities between $a, b : \mathcal{C}O$ consist of
 - $\phi_{x\bullet} : \mathcal{C}A(x, a) \cong \mathcal{C}A(x, b)$ for each $x : \mathcal{C}O$
 - $\phi_{\bullet z} : \mathcal{C}A(a, z) \cong \mathcal{C}A(b, z)$ for each $z : \mathcal{C}O$
 - $\phi_{\bullet\bullet} : \mathcal{C}A(a, a) \cong \mathcal{C}A(b, b)$
 - The following for all appropriate w, x, y, z, f, g, h :

$$\begin{array}{ll}
 \mathcal{C}T_{x,y,a}(f, g, h) \leftrightarrow \mathcal{C}T_{x,y,b}(f, \phi_{y\bullet}(g), \phi_{x\bullet}(h)) & \mathcal{C}I_a(f) \leftrightarrow \mathcal{C}I_b(\phi_{\bullet\bullet}(f)) \\
 \mathcal{C}T_{x,a,z}(f, g, h) \leftrightarrow \mathcal{C}T_{x,b,z}(\phi_{x\bullet}(f), \phi_{\bullet z}(g), h) & \mathcal{C}E_{x,a}(f, g) \leftrightarrow \mathcal{C}E_{x,b}(\phi_{x\bullet}(f), \phi_{x\bullet}(g)) \\
 \mathcal{C}T_{a,z,w}(f, g, h) \leftrightarrow \mathcal{C}T_{b,z,w}(\phi_{\bullet z}(f), g, \phi_{\bullet w}(h)) & \mathcal{C}E_{a,x}(f, g) \leftrightarrow \mathcal{C}E_{b,x}(\phi_{\bullet x}(f), \phi_{\bullet x}(g)) \\
 \mathcal{C}T_{x,a,a}(f, g, h) \leftrightarrow \mathcal{C}T_{x,b,b}(\phi_{x\bullet}(f), \phi_{\bullet\bullet}(g), \phi_{x\bullet}(h)) & \mathcal{C}E_{a,a}(f, g) \leftrightarrow \mathcal{C}E_{b,b}(\phi_{\bullet\bullet}(f), \phi_{\bullet\bullet}(g)) \\
 \mathcal{C}T_{a,x,a}(f, g, h) \leftrightarrow \mathcal{C}T_{b,x,b}(\phi_{\bullet x}(f), \phi_{x\bullet}(g), \phi_{\bullet\bullet}(h)) & \\
 \mathcal{C}T_{a,a,x}(f, g, h) \leftrightarrow \mathcal{C}T_{b,b,x}(\phi_{\bullet\bullet}(f), \phi_{\bullet x}(g), \phi_{\bullet x}(h)) & \\
 \mathcal{C}T_{a,a,a}(f, g, h) \leftrightarrow \mathcal{C}T_{b,b,b}(\phi_{\bullet\bullet}(f), \phi_{\bullet\bullet}(g), \phi_{\bullet\bullet}(h)) &
 \end{array}$$

- But this is an isomorphism in the usual categorical sense.
- Univalence at O means that $a = b$ is equivalent to $a \cong b$.
- cf. complete Segal spaces

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong^*_{\mathcal{L}\text{-Str}} T) \simeq (S \twoheadrightarrow T)$$

where $\cong^*_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}}^* T) \simeq (S \twoheadrightarrow T)$$

where $\cong_{\mathcal{L}\text{-Str}}^*$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

Very surjective morphisms of $\mathcal{L}_{\text{Cat}}$ -structures

A very surjective morphism or equivalence $F : \mathcal{C} \simeq \mathcal{D}$ of $\mathcal{L}_{\text{Cat}}$ -structures consists of surjections

- $FO : \mathcal{C}O \twoheadrightarrow \mathcal{D}O$
- $FA : \mathcal{C}A(x, y) \twoheadrightarrow \mathcal{D}A(Fx, Fy)$ for every $x, y : \mathcal{C}O$
- $FT : \mathcal{C}T(f, g, h) \twoheadrightarrow \mathcal{D}T(Ff, Fg, Fh)$ for all $f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z)$
- $FE : \mathcal{C}E(f, g) \twoheadrightarrow \mathcal{D}E(Ff, Fg)$ for all $f, g : \mathcal{C}A(x, y)$
- $FI : \mathcal{C}I(f) \twoheadrightarrow \mathcal{D}I(Ff)$ for all $f : \mathcal{C}A(x, x)$

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong^*_{\mathcal{L}\text{-Str}} T) \simeq (S \twoheadrightarrow T)$$

where $\cong^*_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

Very surjective morphisms of $\mathcal{L}_{\text{Cat}}$ -structures

A very surjective morphism or equivalence $F : \mathcal{C} \simeq \mathcal{D}$ of **univalent** $\mathcal{L}_{\text{Cat}}$ -structures consists of surjections

- $FO : \mathcal{C}O \twoheadrightarrow \mathcal{D}O$
- $FA : \mathcal{C}A(x, y) \twoheadrightarrow \mathcal{D}A(Fx, Fy)$ for every $x, y : \mathcal{C}O$
- $FT : \mathcal{C}T(f, g, h) \twoheadrightarrow \mathcal{D}T(Ff, Fg, Fh)$ for all $f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z)$
- $FE : \mathcal{C}E(f, g) \twoheadrightarrow \mathcal{D}E(Ff, Fg)$ for all $f, g : \mathcal{C}A(x, y)$
- $FI : \mathcal{C}I(f) \twoheadrightarrow \mathcal{D}I(Ff)$ for all $f : \mathcal{C}A(x, x)$

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong^*_{\mathcal{L}\text{-Str}} T) \simeq (S \twoheadrightarrow T)$$

where $\cong^*_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

Very surjective morphisms of $\mathcal{L}_{\text{Cat}}$ -structures

A very surjective morphism or equivalence $F : \mathcal{C} \simeq \mathcal{D}$ of **univalent** $\mathcal{L}_{\text{Cat}}$ -structures consists of surjections

- $FO : \mathcal{C}O \twoheadrightarrow \mathcal{D}O$
- $FA : \mathcal{C}A(x, y) \twoheadrightarrow \mathcal{D}A(Fx, Fy)$ for every $x, y : \mathcal{C}O$
- $FT : \mathcal{C}T(f, g, h) \longleftrightarrow \mathcal{D}T(Ff, Fg, Fh)$ for all $f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z)$
- $FE : \mathcal{C}E(f, g) \longleftrightarrow \mathcal{D}E(Ff, Fg)$ for all $f, g : \mathcal{C}A(x, y)$
- $FI : \mathcal{C}I(f) \longleftrightarrow \mathcal{D}I(Ff)$ for all $f : \mathcal{C}A(x, x)$

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong^*_{\mathcal{L}\text{-Str}} T) \simeq (S \twoheadrightarrow T)$$

where $\cong^*_{\mathcal{L}\text{-Str}}$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

Very surjective morphisms of $\mathcal{L}_{\text{Cat}}$ -structures

A very surjective morphism or equivalence $F : \mathcal{C} \simeq \mathcal{D}$ of **univalent** $\mathcal{L}_{\text{Cat}}$ -structures consists of surjections

- $FO : \mathcal{C}O \twoheadrightarrow \mathcal{D}O$
- $FA : \mathcal{C}A(x, y) \twoheadrightarrow \mathcal{D}A(Fx, Fy)$ for every $x, y : \mathcal{C}O$
- $FT : \mathcal{C}T(f, g, h) \leftrightarrow \mathcal{D}T(Ff, Fg, Fh)$ for all $f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z)$
- $FE : (f = g) \leftrightarrow (Ff = Fg)$ for all $f, g : \mathcal{C}A(x, y)$
- $FI : \mathcal{C}I(f) \leftrightarrow \mathcal{D}I(Ff)$ for all $f : \mathcal{C}A(x, x)$

The right notion of equivalence

Main theorem

For two *univalent* $\mathcal{L}$ -structures S, T ,

$$(S =_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}} T) \simeq (S \cong_{\mathcal{L}\text{-Str}}^* T) \simeq (S \twoheadrightarrow T)$$

where $\cong_{\mathcal{L}\text{-Str}}^*$ denotes levelwise equivalence up to indiscernibility and $\twoheadrightarrow$ denotes a very split surjective morphism.

Very surjective morphisms of $\mathcal{L}_{\text{Cat}}$ -structures

A very surjective morphism or equivalence $F : \mathcal{C} \simeq \mathcal{D}$ of **univalent** $\mathcal{L}_{\text{Cat}}$ -structures consists of surjections

- $FO : \mathcal{C}O \twoheadrightarrow \mathcal{D}O$
- $FA : \mathcal{C}A(x, y) \cong \mathcal{D}A(Fx, Fy)$ for every $x, y : \mathcal{C}O$
- $FT : \mathcal{C}T(f, g, h) \leftrightarrow \mathcal{D}T(Ff, Fg, Fh)$ for all $f : \mathcal{C}A(x, y), g : \mathcal{C}A(y, z), h : \mathcal{C}A(x, z)$
- $FE : (f = g) \leftrightarrow (Ff = Fg)$ for all $f, g : \mathcal{C}A(x, y)$
- $FI : \mathcal{C}I(f) \leftrightarrow \mathcal{D}I(Ff)$ for all $f : \mathcal{C}A(x, x)$

Summary

For every signature $\mathcal{L}$, we have

- a notion of structure,
- a notion of indiscernibility within each sort,
- a notion of univalent structures,
- a notion of equivalence,
- a univalence theorem.

Summary

For every signature $\mathcal{L}$, we have

- a notion of structure,
- a notion of indiscernibility within each sort,
- a notion of univalent structures,
- a notion of equivalence,
- a univalence theorem.

The paper includes examples of

- $\dagger$ -categories,
- profunctors,
- bicategories,
- opetopic bicategories,
- ...

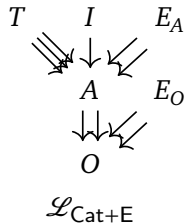
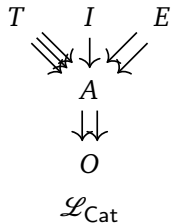
Outline

- 1 The equivalence principle
- 2 Dependent type theory
- 3 Univalent foundations
- 4 Indiscernibility by example
- 5 The univalence principle¹
- 6 Application: double categories²**

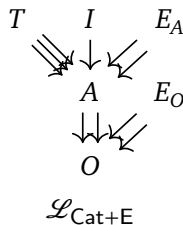
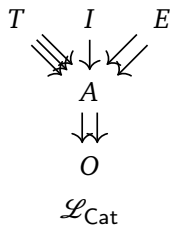
¹jww Ahrens, Shulman, Tsementzis

²jww Rasekh, van der Weide, Ahrens

Categories and set-categories

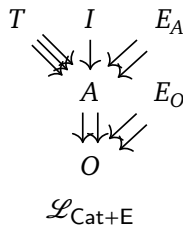
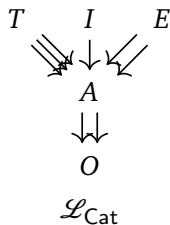


Categories and set-categories



- When we consider $\mathcal{L}_{\text{Cat}+\text{E}}$ -structures (with axioms), the notion of equivalence becomes isomorphism.

Categories and set-categories



- When we consider $\mathcal{L}_{\text{Cat}+\text{E}}$ -structures (with axioms), the notion of equivalence becomes isomorphism.
- Different notions of equivalence are appropriate at different times.

Bicategories

Equivalences for bicategories

- Bicategorical equivalence

Bicategories

Equivalences for bicategories

- Bicategorical equivalence
- Bicategorical equivalence which has isomorphisms of hom-categories

Bicategories

Equivalences for bicategories

- Bicategorical equivalence
- Bicategorical equivalence which has isomorphisms of hom-categories
- Isomorphism

Bicategories

Equivalences for bicategories

- Bicategorical equivalence
- Bicategorical equivalence which has isomorphisms of hom-categories
- Isomorphism

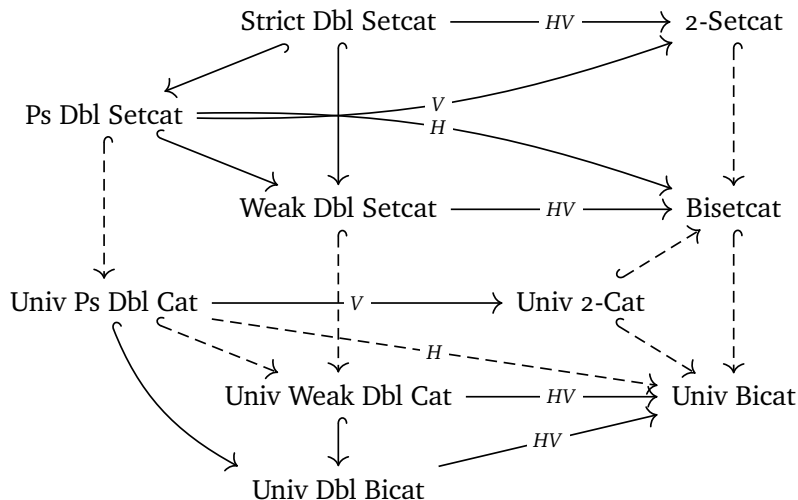
Bicategories

Equivalences for bicategories

- Bicategorical equivalence
- Bicategorical equivalence which has isomorphisms of hom-categories
- Isomorphism

We can give different definitions of bicategory for each.

Double categories, formalized in UniMath



Thank you!