

# The Univalence Principle

Benedikt Ahrens

j.w.w. Paige R North, Michael Shulman, Dimitris Tsementzis

# Contents of this talk

## What this talk is about

1. Quest for an equivalence-invariant foundation of mathematics
2. Overview: how can Voevodsky's Univalent Foundations (UF) be equivalence-invariant?
3. Our work on proving that UF are equivalence-invariant

## What this talk is not about

- Precise definitions and proofs

# Contents of this talk

## What this talk is about

1. Quest for an equivalence-invariant foundation of mathematics
2. Overview: how can Voevodsky's Univalent Foundations (UF) be equivalence-invariant?
3. Our work on proving that UF are equivalence-invariant

## What this talk is not about

- Precise definitions and proofs

For precise mathematical results I refer to our book:

The Univalence Principle (arXiv:2102.06275)

# Overview

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations
- 3 The Univalence Principle
- 4 Example: Univalence Principle for monoids, manually
- 5 Example: Univalence Principle for monoids, in our framework

# Outline

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations
- 3 The Univalence Principle
- 4 Example: Univalence Principle for monoids, manually
- 5 Example: Univalence Principle for monoids, in our framework

# Indiscernibility of identicals

## Indiscernibility of identicals

$$x = y \rightarrow \forall P (P(x) \leftrightarrow P(y))$$

- Reasoning **in logic** is invariant under equality
- **In mathematics**, reasoning should be invariant under weaker notion of sameness!

## Equivalence principle

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

# Aczel's Structure Identity Principle

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

$$G \cong H \rightarrow \forall \text{ group-theoretic properties } P, (P(G) \leftrightarrow P(H))$$

An equivalence principle for category theorists

$$A \simeq B \rightarrow \forall \text{ category-theoretic properties } P, (P(A) \leftrightarrow P(B))$$

Aczel's Structure Identity Principle (SIP)

*Isomorphic mathematical structures are structurally identical; i.e. have the same structural properties.*

# Aczel's Structure Identity Principle

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

$$G \cong H \rightarrow \forall \text{ group-theoretic properties } P, (P(G) \leftrightarrow P(H))$$

An equivalence principle for category theorists

$$A \simeq B \rightarrow \forall \text{ category-theoretic properties } P, (P(A) \leftrightarrow P(B))$$

Aczel's Structure Identity Principle (SIP)

*Isomorphic mathematical structures are structurally identical; i.e. have the same structural properties.*

What are “structural” properties?

# Violating the equivalence principle

What is **not** a structural property?

## Exercise

Find a statement about categories that is not invariant under the equivalence of categories



## Violating the equivalence principle

What is **not** a structural property?

### Exercise

Find a statement about categories that is not invariant under the equivalence of categories



### A solution

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

## Violating the equivalence principle

What is **not** a structural property?

### Exercise

Find a statement about categories that is not invariant under the equivalence of categories



### A solution

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

Can we rule out such “non-structural” statements?

## 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 nonsense.*

## 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 nonsense.*

### Makkai's FOLDS (First Order Logic with Dependent Sorts)

A language for categorical structures in which only invariant properties can be expressed

- FOLDS is not a foundation of mathematics
- Invariance only for properties, not for constructions

# Univalent Foundations and the Univalence Principle

## Voevodsky's goals

- Univalent Foundations as an “invariant language”
- Invariance not only for statements, but also for constructions:  
**any construction on objects in UF can be transported along equivalences of objects**

## Essential ingredients for Univalent Foundations

- Martin-Löf identity type
- Voevodsky's univalence axiom

# Univalent Foundations and the Univalence Principle

## Voevodsky's goals

- Univalent Foundations as an “invariant language”
- Invariance not only for statements, but also for constructions: **any construction on objects in UF can be transported along equivalences of objects**

## Essential ingredients for Univalent Foundations

- Martin-Löf identity type
- Voevodsky's univalence axiom

## In the rest of this talk

How is reasoning in Univalent Foundations invariant under equivalence?

# Outline

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations**
- 3 The Univalence Principle
- 4 Example: Univalence Principle for monoids, manually
- 5 Example: Univalence Principle for monoids, in our framework

## Overview of types in type theory

Type former	Notation	(special case)	canonical term
Inhabitant	$a : A$		
Dependent type	$x : A \vdash B(x)$		
Sigma type	$\sum_{x:A} B(x)$	$A \times B$	$(a, b)$
Product type	$\prod_{x:A} B(x)$	$A \rightarrow B$	$\lambda(x : A).b$
Coproduct type	$A + B$		$\text{inl}(a), \text{inr}(b)$
Identity type	$a = b$		$\text{refl}(a) : a = a$
Universe	$\mathcal{U}$		
Base types	Nat, Bool, 1, 0		

# Identity vs equality

## Inhabitants of $x = y$ behave like equality in many ways

- $\text{refl}(x) : x = x$
- $\text{sym} : x = y \rightarrow y = x$
- $\text{trans} : x = y \times y = z \rightarrow x = z$

## Transport

$$\text{transport} : x = y \rightarrow \prod_{B:A \rightarrow \mathcal{U}} (B(x) \simeq B(y))$$

## Inhabitants of $x = y$ behave **unlike** equality

- Can iterate identity type
- Cannot show that any two identities are identical

# The important features of univalent foundations

## Homotopy levels

- Stratification of types according to “complexity” of their identity types
- Logic: notion of **propositions** given by one layer of this hierarchy

## Univalence axiom

Specifies the identity type of a universe

# Contractible types, propositions and sets

- $A$  is **contractible**

$$\text{isContr}(A) \equiv \sum_{x:A} \prod_{y:A} y = x$$

- $A$  is a **proposition**

$$\text{isProp}(A) \equiv \prod_{x,y:A} x = y$$

- $A$  is a **set**

$$\text{isSet}(A) \equiv \prod_{x,y:A} \text{isProp}(x = y)$$

$$\text{Prop} \equiv \sum_{X:\mathcal{U}} \text{isProp}(X) \quad \text{Set} \equiv \sum_{X:\mathcal{U}} \text{isSet}(X)$$

# Contractible types, propositions and sets

- $A$  is **contractible**

$$\text{isContr}(A) \equiv \sum_{x:A} \prod_{y:A} y = x$$

- $A$  is a **proposition**

$$\text{isProp}(A) \equiv \prod_{x,y:A} \text{isContr}(x = y)$$

- $A$  is a **set**

$$\text{isSet}(A) \equiv \prod_{x,y:A} \text{isProp}(x = y)$$

$$\text{Prop} \equiv \sum_{X:\mathcal{U}} \text{isProp}(X) \quad \text{Set} \equiv \sum_{X:\mathcal{U}} \text{isSet}(X)$$

# Equivalences

## Definition

A map  $f : A \rightarrow B$  is an **equivalence** if it has contractible fibers, i.e.,

$$\text{isequiv}(f) \quad :\equiv \quad \prod_{b:B} \text{isContr} \left( \sum_{a:A} f(a) = b \right)$$

The type of equivalences:

$$A \simeq B \quad :\equiv \quad \sum_{f:A \rightarrow B} \text{isequiv}(f)$$

# Outline

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations
- 3 The Univalence Principle**
- 4 Example: Univalence Principle for monoids, manually
- 5 Example: Univalence Principle for monoids, in our framework

## Different notions of equality

### Synthetic vs. analytic equalities

In MLTT, we always have a **synthetic** equality type between  $a, b : T$

$$a =_T b.$$

Depending on  $T$ , we might have a type of **analytic** equalities

$$a \simeq_T b.$$

**Univalence Principle** for  $T$  and  $\simeq_T$  says that this map is an equivalence

$$(a =_T b) \rightarrow (a \simeq_T b)$$

## Different notions of equality

### Synthetic vs. analytic equalities

In MLTT, we always have a **synthetic** equality type between  $a, b : T$

$$a =_T b.$$

Depending on  $T$ , we might have a type of **analytic** equalities

$$a \simeq_T b.$$

**Univalence Principle** for  $T$  and  $\simeq_T$  says that this map is an equivalence

$$(a =_T b) \rightarrow (a \simeq_T b)$$

Univalence **Axiom**: for  $T = \mathcal{U}$  and  $(X \simeq_{\mathcal{U}} Y) =$  “equivalences  $X \rightarrow Y$ ”:

$$(X =_{\mathcal{U}} Y) \rightarrow (X \simeq_{\mathcal{U}} Y)$$

is an equivalence.

## Transport along equivalence of types

$$x =_{\mathcal{U}} y \rightarrow \prod_{(P:\mathcal{U} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence : } \prod_{(x,y:\mathcal{U})} (x =_{\mathcal{U}} y \simeq x \simeq y)$$

$$x \simeq y \rightarrow \prod_{(P:\mathcal{U} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

## Transport along isomorphism of groups

$$x =_{\text{Grp}} y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence : } \prod_{(x,y:\text{Grp})} (x =_{\text{Grp}} y \simeq x \cong y)$$

$$x \cong y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

## Transport along isomorphism of groups

$$x =_{\text{Grp}} y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence} : \prod_{(x,y:\text{Grp})} (x =_{\text{Grp}} y \simeq x \cong y)$$

$$x \cong y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

### Structure Identity Principle

- One can show  $(x =_{\text{Grp}} y \simeq x \cong y)$ , using univalence for types
- Works similarly for many other structures **built from (types that are) sets**
- See Coquand & Danielsson and HoTT book (Section 9.9)

## Transport along isomorphism of groups

$$x =_{\text{Grp}} y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence} : \prod_{(x,y:\text{Grp})} (x =_{\text{Grp}} y \simeq x \cong y)$$

$$x \cong y \rightarrow \prod_{(P:\text{Grp} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

### Structure Identity Principle

- One can show  $(x =_{\text{Grp}} y \simeq x \cong y)$ , using univalence for types
- Works similarly for many other structures **built from (types that are) sets**
- See Coquand & Danielsson and HoTT book (Section 9.9)

What about things that form a **higher** category, e.g., categories themselves?

## Categories in type theory

A **category**  $\mathcal{C}$  is given by

- a type  $\mathcal{C}_0 : \mathcal{U}$  of **objects**
- for any  $a, b : \mathcal{C}_0$ , a set  $\mathcal{C}(a, b) : \mathcal{U}$  of **morphisms**
- operations: identity & composition

$$1_a : \mathcal{C}(a, a)$$

$$(\circ)_{a,b,c} : \mathcal{C}(b, c) \rightarrow \mathcal{C}(a, b) \rightarrow \mathcal{C}(a, c)$$

- axioms: unitality & associativity

$$1 \circ f = f \quad f \circ 1 = f \quad (h \circ g) \circ f = h \circ (g \circ f)$$

## Categories in type theory

A **category**  $\mathcal{C}$  is given by

- a type  $\mathcal{C}_0 : \mathcal{U}$  of **objects**
- for any  $a, b : \mathcal{C}_0$ , a set  $\mathcal{C}(a, b) : \mathcal{U}$  of **morphisms**
- operations: identity & composition

$$1_a : \mathcal{C}(a, a)$$

$$(\circ)_{a,b,c} : \mathcal{C}(b, c) \rightarrow \mathcal{C}(a, b) \rightarrow \mathcal{C}(a, c)$$

- axioms: unitality & associativity

$$1 \circ f = f \quad f \circ 1 = f \quad (h \circ g) \circ f = h \circ (g \circ f)$$

A **univalent category** is a category  $\mathcal{C}$  such that

$$(a = b) \rightarrow (a \cong b)$$

is an equivalence for all  $a, b : \mathcal{C}_0$ .

# Local univalence implies global univalence

## Theorem (A., Kapulkin, Shulman)

For categories  $A$  and  $B$ , let  $A \simeq B$  denote the type of equivalences from  $A$  to  $B$ . If  $A$  and  $B$  are **univalent**, we have

$$(A =_{\text{uCat}} B) \simeq (A \simeq B).$$

## Transport along equivalence of **univalent** categories

$$x =_{\text{uCat}} y \rightarrow \prod_{(P:\text{uCat} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence : } \prod_{(x,y:\text{uCat})} (x =_{\text{uCat}} y \simeq x \simeq y)$$

$$x \simeq y \rightarrow \prod_{(P:\text{uCat} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

# Transport along equivalence of **univalent** categories

$$x =_{\text{uCat}} y \rightarrow \prod_{(P:\text{uCat} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

$$\text{univalence} : \prod_{(x,y:\text{uCat})} (x =_{\text{uCat}} y \simeq x \simeq y)$$

$$x \simeq y \rightarrow \prod_{(P:\text{uCat} \rightarrow \mathcal{U})} (P(x) \simeq P(y))$$

## Univalence Principle for categories

- Holds only for categories that satisfy themselves a univalence condition: **local univalence implies global univalence**
- Univalent categories are the right notion of categories in Univalent Foundations

## Our work: Univalence Principle

1. Define **signature**, **axiom**, and **theory** for mathematical structures, including higher-categorical ones
2. Given a theory  $\mathcal{T} = (\mathcal{L}, T)$ , define
  - $\mathcal{T}$ -models
  - Univalence of  $\mathcal{T}$ -models
  - Equivalence between  $\mathcal{T}$ -models
3. Prove a univalence result for univalent  $\mathcal{T}$ -models:

$$\text{univalence : } \prod_{(x,y:\text{uMod}\mathcal{T})} ((x =_{\text{uMod}\mathcal{T}} y) \simeq (x \simeq_{\text{uMod}\mathcal{T}} y))$$

## Our work: Univalence Principle

1. Define **signature**, **axiom**, and **theory** for mathematical structures, including higher-categorical ones
2. Given a theory  $\mathcal{T} = (\mathcal{L}, T)$ , define
  - $\mathcal{T}$ -models
  - Univalence of  $\mathcal{T}$ -models
  - Equivalence between  $\mathcal{T}$ -models
3. Prove a univalence result for univalent  $\mathcal{T}$ -models:

$$\text{univalence : } \prod_{(x,y:\text{uMod}\mathcal{T})} ((x =_{\text{uMod}\mathcal{T}} y) \simeq (x \simeq_{\text{uMod}\mathcal{T}} y))$$

### Technical challenge

Define a notion of “isomorphism” (called **indiscernibility**) that

1. works for any signature/theory
2. specializes to categorical isomorphism for the theory of categories

# 1-categorical vs higher-categorical structures

When passing from set-level structures to higher-categorical structures, it **looks** like things get more complicated:

1. What is the role of the “local univalence” condition on categories?
2. Are higher-categorical structures fundamentally different from 1-categorical ones?

## In the rest of the talk

1. Show what the local univalence condition means for 1-categorical structures
2. Our example: monoids

# Outline

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations
- 3 The Univalence Principle
- 4 Example: Univalence Principle for monoids, manually**
- 5 Example: Univalence Principle for monoids, in our framework

# Monoids in type theory

In type theory, a monoid is a tuple  $(M, \mu, e, \alpha, \lambda, \rho)$  where

1.  $M : \mathbf{Set}$
2.  $\mu : M \times M \rightarrow M$
3.  $e : M$
4.  $\alpha : \prod_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5.  $\lambda : \prod_{(a:M)} \mu(e,a) = a$
6.  $\rho : \prod_{(a:M)} \mu(a,e) = a$

# Monoids in type theory

In type theory, a monoid is a tuple  $(M, \mu, e, \alpha, \lambda, \rho)$  where

1.  $M : \mathbf{Set}$
2.  $\mu : M \times M \rightarrow M$
3.  $e : M$
4.  $\alpha : \prod_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5.  $\lambda : \prod_{(a:M)} \mu(e,a) = a$
6.  $\rho : \prod_{(a:M)} \mu(a,e) = a$

Why  $M : \mathbf{Set}$ ?

# Monoids in type theory

In type theory, a monoid is a tuple  $(M, \mu, e, \alpha, \lambda, \rho)$  where

1.  $M : \mathbf{Set}$
2.  $\mu : M \times M \rightarrow M$
3.  $e : M$
4.  $\alpha : \prod_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5.  $\lambda : \prod_{(a:M)} \mu(e,a) = a$
6.  $\rho : \prod_{(a:M)} \mu(a,e) = a$

Why  $M : \mathbf{Set}$ ?

Abstractly, a monoid is a (dependent) pair  $(data, proof)$  where

- *data* is 1.–3.
- *proof* is 4.–6.

## The type of monoids

- We want two monoids  $(data, proof)$  and  $(data', proof')$  to be the same if  $data$  is the same as  $data'$ .
- This is guaranteed when the types of  $proof$  and  $proof'$  are **propositions**.
- This in turn is guaranteed when  $M$  is a **set**.

Summarily:

$$\text{Monoid} \quad :\equiv \quad \sum_{(M:\text{Set})} \sum_{(\mu, e):\text{MonoidStr}(M)} \text{MonoidAxioms}(M, (\mu, e))$$

Can show

$$\text{isProp}(\text{MonoidAxioms}(M, (\mu, e)))$$

## Monoid isomorphisms

Given  $\mathbf{M} \equiv (M, \mu, e, \alpha, \lambda, \rho)$  and  $\mathbf{M}' \equiv (M', \mu', e', \alpha', \lambda', \rho')$ , a **monoid isomorphism** is a bijection  $f : M \cong M'$  preserving  $\mu$  and  $e$ .

## Monoid isomorphisms

Given  $\mathbf{M} \equiv (M, \mu, e, \alpha, \lambda, \rho)$  and  $\mathbf{M}' \equiv (M', \mu', e', \alpha', \lambda', \rho')$ , a **monoid isomorphism** is a bijection  $f : M \cong M'$  preserving  $\mu$  and  $e$ .

$$\begin{aligned} \mathbf{M} = \mathbf{M}' &\simeq (M, \mu, e) = (M', \mu', e') \\ &\simeq \sum_{p: M=M'} (\text{transport}^{Y \mapsto (Y \times Y \rightarrow Y)}(p, \mu) = \mu') \\ &\quad \times (\text{transport}^{Y \mapsto Y}(p, e) = e') \\ &\simeq \sum_{f: M \cong M'} (f \circ \mu \circ (f^{-1} \times f^{-1}) = \mu') \\ &\quad \times (f \circ e = e') \\ &\simeq \mathbf{M} \cong \mathbf{M}' \end{aligned}$$

# Transport along monoid isomorphism

We now have two ingredients:

1.

$$\text{transport}_{\mathbf{M}, \mathbf{M}'} : (\mathbf{M} = \mathbf{M}') \rightarrow \prod_{B: \text{Monoid} \rightarrow \mathcal{U}} (B(\mathbf{M}) \simeq B(\mathbf{M}'))$$

2.

$$(\mathbf{M} = \mathbf{M}') \simeq (\mathbf{M} \cong \mathbf{M}')$$

Composing these, we get

$$\text{transport}_{\mathbf{M}, \mathbf{M}'} : (\mathbf{M} \cong \mathbf{M}') \rightarrow \prod_{B: \text{Monoid} \rightarrow \mathcal{U}} (B(\mathbf{M}) \simeq B(\mathbf{M}'))$$

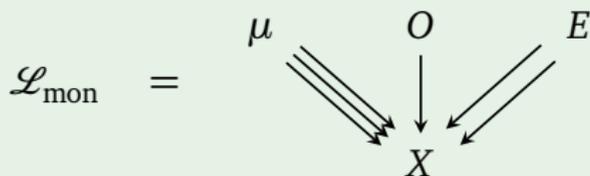
# Outline

- 1 Motivation for Univalent Foundations
- 2 Reminder: Univalent Foundations
- 3 The Univalence Principle
- 4 Example: Univalence Principle for monoids, manually
- 5 Example: Univalence Principle for monoids, in our framework**

# The signature of monoids

## Example

Signature  $\mathcal{L}_{\text{mon}}$  for a monoid:



A **structure**  $M$  for this signature consists of

1. a type  $MX$
2. a family of types  $M\mu(x,y,z)$  for  $x,y,z : MX$
3. a family of types  $MO(x)$  for  $x : MX$
4. a family of types  $ME(x,y)$  for  $x,y : MX$

# The theory of monoids

Not all structures represent monoids. **Axioms** specify those structures that are a monoid:

## Axioms of a monoid

### 1. Monoid axioms:

$$\forall(x, y, z, z' : X). \mu(x, y, z) \rightarrow \mu(x, y, z') \rightarrow E(z, z')$$

$$\forall(x, y : X). \exists(z : X). \mu(x, y, z)$$

$$\forall(x, x', y, z : X). E(x, x') \rightarrow \mu(x, y, z) \rightarrow \mu(x', y, z)$$

$$\forall(x, y : X). E(x, y) \rightarrow E(y, x)$$

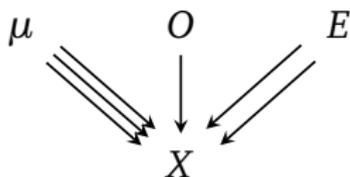
...

### 2. “Homotopical axioms”:

2.1  $MX$  is a set

2.2  $M\mu(x, y, z)$ ,  $MO(x)$ ,  $ME(x, y)$  are pointwise propositions

## Indiscernibility for elements of a monoid



Given  $a, b : MX$ , an **indiscernibility**  $a \asymp b$  consists of “equivalences of types of everything above  $a$  and  $b$ ”

$$M\mu(a, y, z) \simeq M\mu(b, y, z)$$

$$M\mu(x, a, z) \simeq M\mu(x, b, z)$$

$$M\mu(x, y, a) \simeq M\mu(x, y, b)$$

$$M\mu(a, a, z) \simeq M\mu(b, b, z)$$

...

$$MO(a) \simeq MO(b)$$

$$ME(a, y) \simeq ME(b, y)$$

...

# Indiscernibility

1.  $a \asymp b$  means that  $a$  and  $b$  behave in the same way within the structure.
2. In a model  $M$  of the theory of monoids,  $a \asymp b$  reduces to  $ME(a, b)$ .
3. Definition of indiscernibility carries over to any  $\mathcal{L}$ , and any sort in  $\mathcal{L}$ .

# Indiscernibility

1.  $a \asymp b$  means that  $a$  and  $b$  behave in the same way within the structure.
2. In a model  $M$  of the theory of monoids,  $a \asymp b$  reduces to  $ME(a, b)$ .
3. Definition of indiscernibility carries over to any  $\mathcal{L}$ , and any sort in  $\mathcal{L}$ .

## Definition

1. Given  $w, w' : M\mu(a, b, c)$ , an indiscernibility  $w \asymp w'$  is given by an equivalence

$$\mathbf{1} \simeq \mathbf{1}$$

(since there is nothing above  $\mu$  in  $\mathcal{L}_{\text{mon}}$ ). Hence  $(w \asymp w') = \mathbf{1}$ .

2. Similar for  $w, w' : MO(a)$ , and  $w, w' : ME(a, b)$ .

# Univalence of models

## Definition

A monoid  $M$  is **univalent** if the maps

$$(a = b) \rightarrow (a \simeq b) \tag{1}$$

$$(w = w') \rightarrow (w \simeq w') \tag{2}$$

are equivalences.

# Univalence of models

## Definition

A monoid  $M$  is **univalent** if the maps

$$(a = b) \rightarrow (a \simeq b) \tag{1}$$

$$(w = w') \rightarrow (w \simeq w') \tag{2}$$

are equivalences.

## Observations

1. Since  $(w \simeq w') = 1$ , condition (2) is equivalent to  $M\mu$ ,  $MO$ , and  $ME$  being pointwise **propositions**.
2. Since  $(a \simeq b) = ME(a, b)$ , condition (1) is equivalent to  $M$  being a **set with identity**  $a = b$  given by  $E(a, b)$ .

# Univalence of models

## Definition

A monoid  $M$  is **univalent** if the maps

$$(a = b) \rightarrow (a \simeq b) \tag{1}$$

$$(w = w') \rightarrow (w \simeq w') \tag{2}$$

are equivalences.

## Observations

1. Since  $(w \simeq w') = 1$ , condition (2) is equivalent to  $M\mu$ ,  $MO$ , and  $ME$  being pointwise **propositions**.
2. Since  $(a \simeq b) = ME(a, b)$ , condition (1) is equivalent to  $M$  being a **set with identity**  $a = b$  **given by**  $E(a, b)$ .

Summary: a univalent monoid in this sense is exactly the same a monoid as previously defined.

## Equivalence of models

Given monoids  $M, N$ , an **equivalence** is

$$\begin{aligned} e_X &: MX \rightarrow NX \\ e_\mu &: \prod_{x,y,z:MX} M\mu(x,y,z) \rightarrow M\mu(ex, ey, ez) \\ e_O &: \prod_{x:MX} MO(x) \rightarrow MO(ex) \\ e_E &: \prod_{x,y:MX} ME(x,y) \rightarrow ME(ex, ey) \end{aligned}$$

such that  $e_X$ ,  $(e_\mu)_{x,y,z}$ ,  $(e_O)_x$ , and  $(e_E)_{x,y}$  are (split-)surjective.

### Observations

1. For univalent monoids, condition of  $e_E$  being split-surjective entails that  $e_X$  is injective.
2. An equivalence of univalent monoids is an isomorphism of sets that preserves multiplication and unit.

# Univalence for univalent monoids

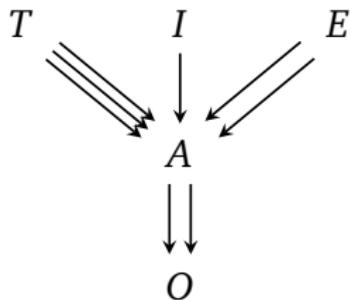
## Theorem (Univalence Principle)

*For univalent  $M$  and  $N$ ,*

$$(M = N) \simeq (M \simeq N)$$

## Local univalence condition on categories

For the signature of categories,



an indiscernibility  $a \simeq b$  in  $MO$  is the same as an isomorphism  $a \cong b$ , via a Yoneda-style argument.

The univalence condition at  $A$  says that  $A$  is a set with  $f = g$  given by  $ME(f, g)$ .

## Other examples

- First-order logic (with equality)
- Higher-order logic, e.g., topological spaces, suplattices
- Categories
- Dagger categories
- (Ana)functors
- Profunctors
- Displayed categories / Fibrations
- Bicategories
- Double categories
- ...

## Where does our work take place?

Working in **two-level type theory** of Annenkov, Capriotti, Kraus, Sattler.

- Univalent Foundations, embedded in an extensional type theory
- Universes  $\mathcal{U} \hookrightarrow \mathcal{U}^s$
- $\mathcal{U}$  implements univalent type theory.
- Every type  $T : \mathcal{U}^s$  is equipped with a strict equality type  $a \equiv_T b$  with the usual rules for the identity type, but which also satisfies UIP.
- Signatures live in  $\mathcal{U}^s$  (are meta-mathematical), but models and their morphisms live in  $\mathcal{U}$  (are mathematical)

## References

- Coquand, Danielsson, “Isomorphism is equality”
- The HoTT book (Section 9.9 for Structure Identity Principle)
- Ahrens, Kapulkin, Shulman, “Univalent categories and the Rezk completion”
- Ahrens, North, Shulman, Tsementzis, “The Univalence Principle”