The Univalence Principle

Benedikt Ahrens

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

Interactions of Proof Assistants and Mathematics
International Summer School, Regensburg, Germany
2023-09-26

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

Outline

Indiscernibility of identicals

Indiscernibility of identicals

$$x = y \quad \to \quad \forall P\, (P(x) \leftrightarrow P(y))$$

- 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**.

Invariance under sameness

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

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

An equivalence principle for category theorists

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

Invariance under sameness

Notion of sameness depends on the objects under consideration:

An equivalence principle for group theorists

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

An equivalence principle for category theorists

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

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 $\mathscr{C}$ 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 $\mathscr{C}$ 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

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	$\prod_{x:A} B(x)$	$A \to B$	$\lambda(x : A).b$
Coproduct type	$A + B$		$\mathsf{inl}(a)$, $\mathsf{inr}(b)$
Identity type	$a = b$		$\mathsf{refl}(a) : a = a$
Universe	$\mathcal{U}$		
Base types	Nat, Bool, $\mathsf{1}$, $\mathsf{0}$		

Identity vs equality

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

- $\mathsf{refl}(x) : x = x$
- $\mathsf{sym}(x, y) : x = y \to y = x$
- $\mathsf{trans}(x, y, z) : x = y \times y = z \to x = z$

Transport

$$\mathsf{transport} : x = y \to \prod_{B : A \to \mathcal{U}} \big(B(x) \simeq B(y)\big)$$

Inhabitants of $x = y$ behave **un**like 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**

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

- A is a **proposition**

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

- A is a **set**

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

$$\mathsf{Prop} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isProp}(X) \qquad \mathsf{Set} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isSet}(X)$$

Contractible types, propositions and sets

- A is **contractible**

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

- A is a **proposition**

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

- A is a **set**

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

$$\mathsf{Prop} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isProp}(X) \qquad \mathsf{Set} \;:\equiv\; \sum_{X:\mathcal{U}} \mathsf{isSet}(X)$$

Equivalences

Definition

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

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

The type of equivalences:

$$A \simeq B \;:\equiv\; \sum_{f:A\to B} \mathsf{isequiv}(f)$$

Outline

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

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

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

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

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

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

is an equivalence.

Transport along equivalence of types

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

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

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

Transport along biimplication

$$(P = Q) \to \prod_{(S:\mathsf{Prop}\to\mathcal{U})} (S(P) \simeq S(Q))$$

$$\mathsf{univalence} : \prod_{(P,Q:\mathsf{Prop})} ((P = Q) \xrightarrow{\simeq} (P \leftrightarrow Q))$$

$$(P \leftrightarrow Q) \to \prod_{(S:\mathsf{Prop}\to\mathcal{U})} (S(P) \simeq S(Q))$$

Transport along bijections

$$(X = Y) \to \prod_{(S:\mathsf{Set}\to\mathcal{U})} (P(X) \simeq P(Y))$$

$$\mathsf{univalence} : \prod_{(X,Y:\mathsf{Set})} ((X = Y) \xrightarrow{\simeq} (X \cong Y))$$

$$(X \cong Y) \to \prod_{(S:\mathsf{Set}\to\mathcal{U})} (P(X) \simeq P(Y))$$

Transport along isomorphism of groups

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

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

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

Transport along isomorphism of groups

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

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

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

Structure Identity Principle

- One can show $(x =_{\mathsf{Grp}} y \xrightarrow{\simeq} x \cong y)$, 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

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

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

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

Structure Identity Principle

- One can show $(x =_{\mathsf{Grp}} y \xrightarrow{\sim} x \cong y)$, 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** $\mathscr{C}$ is given by

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

$$\begin{aligned} 1_a &: \mathscr{C}(a,a) \\ (\circ)_{a,b,c} &: \mathscr{C}(b,c) \to \mathscr{C}(a,b) \to \mathscr{C}(a,c) \end{aligned}$$

- axioms: unitality & associativity

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

Categories in type theory

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

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

$$\mathrm{I}_a : \mathscr{C}(a,a)$$
$$(\circ)_{a,b,c} : \mathscr{C}(b,c) \to \mathscr{C}(a,b) \to \mathscr{C}(a,c)$$

- axioms: unitality & associativity

$$\mathrm{I} \circ f = f \qquad f \circ \mathrm{I} = f \qquad (h \circ g) \circ f = h \circ (g \circ f)$$

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

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

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

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*

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

Transport along equivalence of **univalent** categories

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

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

$$x \simeq y \to \prod_{(P:\mathsf{uCat}\to\mathcal{U})} (P(x) \simeq P(y))$$

Transport along equivalence of **univalent** categories

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

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

$$x \simeq y \to \prod_{(P:\mathsf{uCat}\to\mathcal{U})} (P(x) \simeq P(y))$$

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 $\mathscr{T} = (\mathscr{L}, T)$, define
   - $\mathscr{T}$-models
   - Univalence of $\mathscr{T}$-models
   - Equivalence between $\mathscr{T}$-models
3. Prove a univalence result for univalent $\mathscr{T}$-models:

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{uMod}\mathscr{T})} ((x =_{\mathsf{uMod}\mathscr{T}} y) \xrightarrow{\sim} (x \simeq_{\mathsf{uMod}\mathscr{T}} y))$$

Our work: Univalence Principle

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

$$\mathsf{univalence} : \prod_{(x,y:\mathsf{uMod}\mathscr{T})} ((x =_{\mathsf{uMod}\mathscr{T}} y) \xrightarrow{\sim} (x \simeq_{\mathsf{uMod}\mathscr{T}} y))$$

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 higer-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

Monoids in type theory

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

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

Monoids in type theory

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

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

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. $\mu : M \times M \to M$
3. $e : M$
4. $\alpha : \Pi_{(a,b,c:M)} \mu(\mu(a,b),c) = \mu(a,\mu(b,c))$
5. $\lambda : \Pi_{(a:M)} \mu(e,a) = a$
6. $\rho : \Pi_{(a:M)} \mu(a,e) = a$

Why $M : \mathbf{Set}$?

Abstractly, a monoid is a (dependent) pair $(\mathit{data}, \mathit{proof})$ where

- *data* is 1.–3.
- *proof* is 4.–6.

The type of monoids

- We want two monoids $(\mathit{data}, \mathit{proof})$ and $(\mathit{data}', \mathit{proof}')$ 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:

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

Can show

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

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 μ 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 μ and e.

$$\begin{aligned}
\mathbf{M} = \mathbf{M}' &\simeq (M, \mu, e) = (M', \mu', e') \\
&\simeq \sum_{p:M=M'} (\mathsf{transport}^{Y \mapsto (Y \times Y \to Y)}(p, \mu) = \mu') \\
&\qquad\qquad \times (\mathsf{transport}^{Y \mapsto Y}(p, e) = e') \\
&\simeq \sum_{f:M \cong M'} \left(f \circ \mu \circ (f^{-1} \times f^{-1}) = \mu'\right) \\
&\qquad\qquad \times (f \circ e = e') \\
&\simeq \mathbf{M} \cong \mathbf{M}'
\end{aligned}$$

Transport along monoid isomorphism

We now have two ingredients:

1.
$$\mathbf{transport}_{\mathbf{M},\mathbf{M}'} : (\mathbf{M} = \mathbf{M}') \to \prod_{B:\mathbf{Monoid}\to\mathcal{U}} \big(B(\mathbf{M}) \simeq B(\mathbf{M}')\big)$$

2.
$$(\mathbf{M} = \mathbf{M}') \;\simeq\; (\mathbf{M} \cong \mathbf{M}')$$

Composing these, we get

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

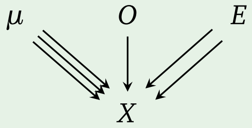
Outline

The signature of monoids

Example

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

$$\mathscr{L}_{\text{mon}} \quad =$$



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:

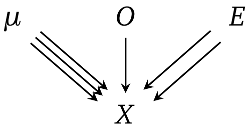
Axioms of a monoid

1. Monoid axioms:

$$\forall(x,y,z,z' : X).\mu(x,y,z) \to \mu(x,y,z') \to E(z,z')$$
$$\forall(x,y : X).\exists(z : X).\mu(x,y,z)$$
$$\forall(x,x',y,z : X).E(x,x') \to \mu(x,y,z) \to \mu(x',y,z)$$
$$\forall(x,y : X).E(x,y) \to E(y,x)$$
$$\ldots$$

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"

$$M\mu(a, y, z) \simeq M\mu(b, y, z)$$
$$M\mu(x, a, z) \simeq M\mu(x, b, z)$$
$$M\mu(x, y, a) \simeq M\mu(x, y, b)$$
$$M\mu(a, a, z) \simeq M\mu(b, b, z)$$
$$\dots$$
$$MO(a) \simeq MO(b)$$
$$ME(a, y) \simeq ME(b, y)$$
$$\dots$$

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 $\mathscr{L}$, and any sort in $\mathscr{L}$.

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 $\mathscr{L}$, and any sort in $\mathscr{L}$.

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 μ in $\mathscr{L}_{\text{mon}}$). 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

$$(a = b) \to (a \asymp b) \tag{1}$$
$$(w = w') \to (w \asymp w') \tag{2}$$

are equivalences.

Univalence of models

Definition

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

$$(a = b) \to (a \asymp b) \tag{1}$$

$$(w = w') \to (w \asymp w') \tag{2}$$

are equivalences.

Observations

1. Since $(w \asymp w') = 1$, condition (2) is equivalent to $M\mu$, MO, and ME being pointwise **propositions**.
2. Since $(a \asymp b) = ME(a, b)$, 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

$$(a = b) \to (a \asymp b) \tag{1}$$

$$(w = w') \to (w \asymp w') \tag{2}$$

are equivalences.

Observations

1. Since $(w \asymp w') = 1$, condition (2) is equivalent to $M\mu$, MO, and ME being pointwise **propositions**.
2. Since $(a \asymp b) = ME(a, b)$, 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

$$e_X : MX \to NX$$

$$e_\mu : \prod_{x,y,z:MX} M\mu(x,y,z) \to N\mu(ex,ey,ez)$$

$$e_O : \prod_{x:MX} MO(x) \to NO(ex)$$

$$e_E : \prod_{x,y:MX} : ME(x,y) \to NE(ex,ey)$$

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

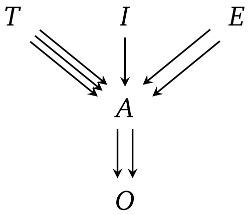
Theorem (Univalence Principle)

For univalent M and N,

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

Local univalence condition on categories

For the signature of categories,



an indiscernibility $a \asymp b$ 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 $ME(f, g)$.

Outline

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 $\mathscr{U} \hookrightarrow \mathscr{U}^s$
- $\mathscr{U}$ implements univalent type theory.
- Every type $T : \mathscr{U}^s$ 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 $\mathscr{U}^s$ (are meta-mathematical), but models and their morphisms live in $\mathscr{U}$ (are mathematical)

Signatures

- Signatures are abstract specification devices for mathematical structures
- We have two notions of signature.

“Diagram” Signatures

- Certain categories where
  - objects indicate sorts
  - morphisms indicate dependencies

+ Intuitive

– Complicated to reason about

“Functorial” Signatures

- (Co)inductively defined

+ Easy to reason about

– Difficult/unintuitive for specifying instances

Translation of Signatures

- We use diagram signatures for examples
- All results are proved for functorial signatures
- Algorithmic translation from diagram to functorial signatures
- Functorial signatures axiomatize the operation of “derivation”
- Functorial signatures are more general than diagram signatures

Diagram Signatures and Their Models

Definition (Diagram Signatures)

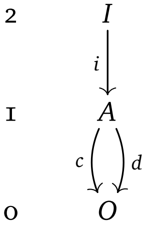
Too complicated, let's just draw some examples!

Definition (Models of a Diagram Signature)

Even more complicated. . .

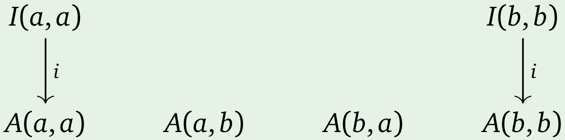
- Definition of model uses "derivation" of signatures
- Derivation does not preserve "finiteness" of signatures

Example of derivation



Example

In $\mathscr{L}_{\mathrm{rg}}$ we have $(\mathscr{L}_{\mathrm{rg}})_0 \equiv \{O\}$. Let M_0 be a (a function picking out) the two-element set $\{a, b\}$. Then $(\mathscr{L}_{\mathrm{rg}})'_{M_0}$ is the following signature, with four sorts of rank 0 and two sorts of rank 1:



Functorial Signatures

Observation: Essential Features of a Diagram Signature $\mathscr{L}$

- The type $\mathscr{L}(\mathrm{o})$ of non-dependent sorts
- The derived diagram signature $\mathscr{L}'_M$ for any $M : \mathscr{L}_{\mathrm{o}} \to \mathscr{U}$.

Definition (Functorial Signature, coinductively)

Consists of

- a type $\mathscr{L}_{\mathrm{o}}$
- for any $M : L_{\mathrm{o}} \to \mathscr{U}$, a functorial signature $\mathscr{L}'(M)$

- Definition can be made inductive by decorating it with a decreasing **height**.
- Need to define not just the pretype, but the strict **category** of functorial signatures (of height n).

Models of a Functorial Signature

Definition ($\mathscr{L}$-structure)

- Of $\mathscr{L}$ of height 0: a unique structure
- Of $\mathscr{L}$ of height $n+1$:
  1. A function $M : \mathscr{L}_0 \to \mathscr{U}$
  2. a structure of $\mathscr{L}'(M)$.

- Morphism of structures, composition, identity
- Pullback of structures along morphisms of signatures

Axioms and Theories

- An $\mathscr{L}$**-axiom** is a function $\mathbf{Struc}(\mathscr{L}) \to \mathsf{hProp}$.
- A **functorial theory** is a pair $(\mathscr{L}, T)$ of a functorial signature $\mathscr{L}$ and a family T of $\mathscr{L}$-axioms

Outline

Indiscernibility

Goal

Define a notion of "isomorphism" for elements of a structure

Definition (Indiscernibility)

Too complicated to write down, will be defined for structures of diagram signatures by example.

Definition (Univalence)

An $\mathscr{L}$-structure M is **univalent** if for any sort K of $\mathscr{L}$ and any $a, b : M(K)$, the map

$$a = b \to a \cong b$$

is an equivalence.

Results on Homotopy Levels

Theorem

If $\mathscr{L}$ has height $n+1$, $M : \mathbf{Struc}(\mathscr{L})$ is univalent, and $K : \mathscr{L}(0)$, then MK is an $(n-1)$-type.

Theorem

If $\mathscr{L}$ has height n, then the type of univalent $\mathscr{L}$-structures is an $(n-1)$-type.

Univalence Principle

Theorem

For any functorial signature $\mathscr{L}$ and $M, N : \mathbf{Struc}(\mathscr{L})$ that are both univalent, the canonical map

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

is an equivalence.

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
- ...

Outline

Higher-Order Logic: Topological Spaces

- $M \mapsto (M \to \mathsf{hProp}) \to \mathsf{hProp}$
- Space M is such a structure by equipping it with the family of all supersets of the set of open subsets, i.e., a predicate that holds of X just when $U \in X$ for every open subset U of M.
- Morphism of structures: $f : M \to N$ such that if X contains all opens in M, then its image under f contains all opens in N, which is to say that $f^{-1}(U) \in X$ for all opens U in N. This is equivalent to saying that $f^{-1}(U)$ is open in M for all opens U in N, i.e., that f is continuous.

Suplattices

- $$\mathcal{L}'_M =_{\text{df}} (M \times M) + ((\textstyle\sum_{(A:\mathbf{Set})}(A \to M)) \times M)$$
- $M \times M$ stands for the partial ordering—(m, n) meaning $m \leq n$—whereas the second summand denotes suprema: (X, s) holds if and only if s is a supremum of the family X of elements of M.
- Structure M, then $m_1, m_2 : M$ are indiscernible if $m_1 \leq m_2$ and $m_2 \leq m_1$. (That m_1 and m_2 are suprema of exactly the same families X is then automatic.)
- Univalence at bottom level means that M is a set, and that the preorder $\leq$ on M is antisymmetric.
- Morphism of structures is sup-preserving morphism of preorders
- It is an equivalence if it is (split) surjective up to isomorphism and reflects the preorder (and hence also suprema of families).

Open Questions

- Completion operation for structures — turning a structure into a univalent one, universally? Ongoing work, e.g., by Kobe Wullaert.
- Univalence principle for structures of infinite height?
- Class of axioms that are invariant under weak equivalence?
- Signatures where functions are native, that is, not expressed as functional relations?
- Formalization? Ongoing work, e.g., by Elif Uskuplu.

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”