Univalent foundations and UniMath for the formalization of (higher) category theory

Benedikt Ahrens

Machine-Assisted Proofs 2023
IPAM, UCLA, CA, USA

2023-02-13

Programme for this talk

Aspects of Machine-Assisted Proofs

- Numerical calculation
- Symbolic reasoning
- Logic/foundations
- Artificial Intelligence
- Machine Learning
- User Interfaces
- ...

My talk

Univalent foundations as a suitable (domain-specific) foundation for (computer) formalization of category theory

What is special about category theory?

- Category theory is about **sameness** (isomorphism) of mathematical objects
- Categories themselves form something more complicated than a category; sameness of categories is not isomorphism, but equivalence
- Higher categories have even looser notions of sameness
- Univalent foundations allow one to express reasoning exactly modulo this kind of sameness

Outline

Outline

Indiscernibility of identicals

Indiscernibility of identicals

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

- 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**.

Equivalence Principle is domain-specific

An equivalence principle for groups

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

An equivalence principle for categories

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

Equivalence Principle is domain-specific

An equivalence principle for groups

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

An equivalence principle for categories

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

What are “structural” properties?

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a property of categories that is not invariant under the equivalence of categories

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a property of categories that is not invariant under the equivalence of categories

A solution

The property of a category of having exactly one object

Violating the equivalence principle

What is **not** a structural property?

Exercise

Find a property of categories that is not invariant under the equivalence of categories

A solution

The property of a category of having exactly one object

How can we identify “non-structural” statements?

Reasoning modulo equivalence

A mathematician's view

- Understands intuitively if a given statement is invariant under equivalence

Why make it more explicit?

- Might want to transfer constructions, not just proofs.
- For complicated mathematical objects, equivalences are complicated.
- Computer proof assistants require all the details.

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.*

Univalent Foundations and the Univalence Principle

Vladimir 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**

Univalent Foundations and the Univalence Principle

Vladimir 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**

[. . .] My homotopy lambda calculus is an attempt to create a system which is very good at dealing with equivalences. In particular it is supposed to have the property that given any type expression F(T) depending on a term subexpression t of type T and an equivalence t->t' (a term of the type Eq(T;t,t')) there is a mechanical way to create a new expression F' now depending on t' and an equivalence between F(T) and F'(T') (note that to get F' one can not just substitute t' for t in F – the resulting expression will most likely be syntactically incorrect). *Email from VV to Dan Grayson, Sept 2006*

Equivalence Principle in Univalent Foundations

In univalent foundations, we can show

An equivalence principle for groups

$$(G \cong H) \quad \to \quad \forall P, (P(G) \leftrightarrow P(H))$$

An equivalence principle for categories

$$(A \simeq B) \quad \to \quad \forall P, (P(A) \leftrightarrow P(B))$$

Outline

Martin-Löf type theory by example

- Dependent types:

$$\begin{aligned} n : \mathsf{Nat} &\vdash \mathsf{IsEven}(n) \\ X : \mathscr{U} &\vdash \mathsf{GroupStructure}(X) \end{aligned}$$

- For any natural number n, $n + n$ is even:

$$\prod_{(n:\mathsf{Nat})} \sum_{(k:\mathsf{Nat})} n + n = 2 * k$$

- Definition of monoid:

$$\sum_{M:\mathscr{U}} \sum_{\mu:M\times M\to M} \sum_{e:M} \left(\prod_{m:M} \mu(m,e) = m \right) \times (\ldots) \times (\ldots)$$

Identity vs equality

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

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

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: $p =_{a=b} q$
- Cannot show that any two identities are identical
- Identity $X =_{\mathcal{U}} Y$ well-formed, but its inhabitants not determined

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

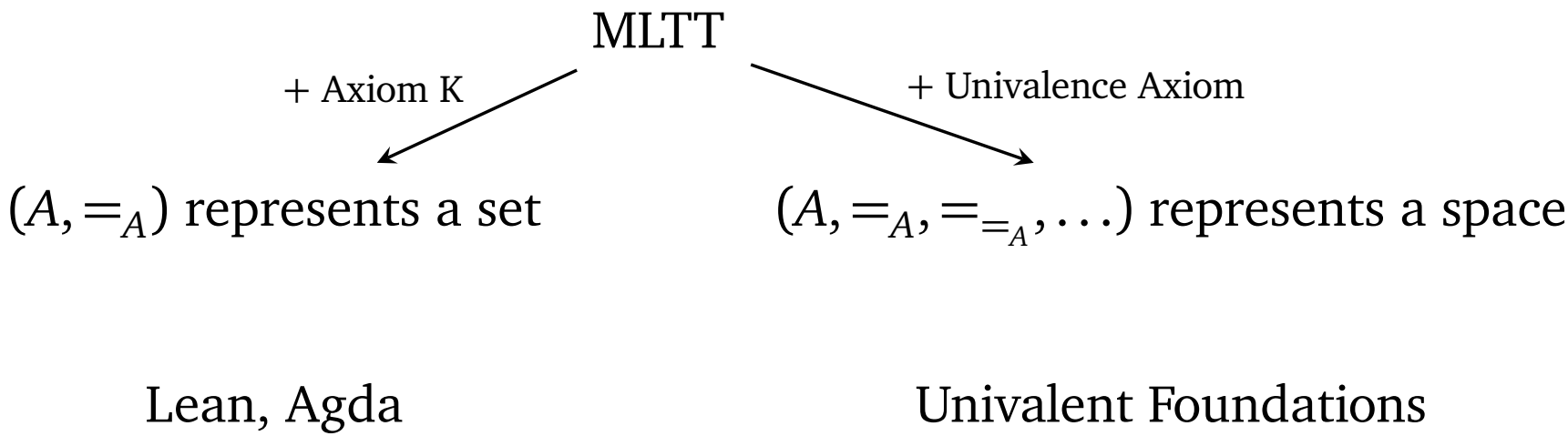
Univalence axiom

Specifies the identity type of a universe:

$$(X =_{\mathcal{U}} Y) \to (X \simeq Y)$$
$$\mathsf{refl}(X) \mapsto \mathrm{id}_X$$

is an equivalence

Univalent foundations as an extension of Martin-Löf type theory



On univalent foundations

- Mathematics is the study of structures on sets and their higher analogs.
- Set-theoretic mathematics constitutes a subset of the mathematics that can be expressed in univalent foundations.
- Classical mathematics is a subset of univalent mathematics consisting of the results that require LEM and/or AC among their assumptions.

see Voevodsky, Talk at HLF, Sept 2016

Voevodsky learning how to use the proof assistant Coq

> *I am thinking a lot these days about foundations of math and automated proof verification. My old idea about a "univalent" homotopy theoretical models of Martin-Lof type systems survived the verification stage an I am in the process of writing things up.*
>
> *I also took a course at the Princeton CS department which was for most part about Coq and was very impressed both by how much can be proved in it in a reasonable time and by how many young students attended (45, 35 undergrad + 10 grad!).*

Email from Vladimir Voevodsky to Dan Grayson, Dec 2009

The Foundations library of computer-checked mathematics

In Feb 2010, Voevodsky started writing the Coq library *Foundations*, making precise his ideas conceived during three years and collected in *A very short note on homotopy λ-calculus*.

```
Fixpoint isofhlevel (n:nat) (X:UU): UU:=
match n with
O => iscontr X |
S m => forall x:X, forall x':X, (isofhlevel m (paths _ x x'))
end.


Theorem hlevelretract (n:nat)(X:UU)(Y:UU)(p:X -> Y)(s:Y ->X)(eps: forall y:Y, paths _  (p (s y)) y): (isofhlevel n X) -> (isofhlevel n Y).
Proof. intro. induction n. intros. apply (contrl1' _ _ p s eps X0).
intros. unfold isofhlevel. intros. unfold isofhlevel in X0. assert (is: isofhlevel n (paths _ (s x) (s x'))).  apply X0.
set (s':= maponpaths _ _ s x x'). set (p':= pathssec2 _ _ s p eps x x'). set (eps':= pathssec3 _ _ s p eps x x'). apply (IHn _ _ p' s' eps' is). Defined.

Corollary  hlevelweqf (n:nat)(X:UU)(Y:UU)(f:X -> Y)(is: isweq _ _ f): (isofhlevel n X) -> (isofhlevel n Y).
Proof. intros.  apply (hlevelretract n _ _ f (invmap _ _ f is) (weqfg _ _ f is)). assumption. Defined.

Corollary  hlevelweqb (n:nat)(X:UU)(Y:UU)(f:X -> Y)(is: isweq _ _ f): (isofhlevel n Y) -> (isofhlevel n X).
Proof. intros.  apply (hlevelretract n _ _ (invmap _ _ f is) f (weqgf _ _ f is)). assumption. Defined.


Definition isaprop (X:UU): UU := isofhlevel (S O) X.
```

This library, and other libraries built on top of *Foundations*, were later combined into the UniMath (Univalent Mathematics) library, which continues to be developed.

Outline

Outline

What is category theory?

- Category theory is a general theory of mathematical structures and their relations
- Originally developed to conceptualize constructions in algebraic topology
- Now used in all sorts of mathematics, in computer science, biology, . . . (see Applied Category Theory conference)

Dimensions

- Notions of category are defined for any "dimension".
- I will only talk about 1-categories, but most of what I say holds for higher categories.

Why formalize category theory at all?

Generic reasons

- Machine-readable storage of mathematical knowledge for archiving and searching
- For use in teaching

Reasons specific(?) to category theory

- (Higher) category theory is complex, structures are huge
  $\rightsquigarrow$ It is easy to make mistakes
- Non-experts are increasingly using category theory
  $\rightsquigarrow$ Formal libraries can make mathematical reasoning accessible to them
- Graphical tools for reasoning in higher categories exist, but the correctness of these tools is not mechanically verified itself
  $\rightsquigarrow$ Connect graphical tools to formalized library, with translation between the two?

What is a category?

A category consists of

- a collection of **objects**
- for objects a and b, a collection of **morphisms** from a to b
- composition of **composable** morphisms
- composition satisfies suitable laws

Examples

- Sets and functions
- Groups and homomorphisms
- (Models of) programming languages and translations
- ...

What is a category?

A category consists of

- a collection of **objects**
- for objects a and b, a collection of **morphisms** from a to b
- composition of **composable** morphisms
- composition satisfies suitable laws

Examples

- Sets and functions
- Groups and homomorphisms
- (Models of) programming languages and translations
- ...

This talk

is about the meaning of the word "collection" above...

Outline

Definition of category, set-based

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

- a **set** $\mathscr{C}_{\mathrm{o}} : \mathscr{U}$ of objects
- for any $a, b : \mathscr{C}_{\mathrm{o}}$, 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

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

Definition of category, space-based

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

- a **space** $\mathscr{C}_0 : \mathscr{U}$ of objects
- for any $a, b : \mathscr{C}_0$, a **space** $\mathscr{C}(a,b) : \mathscr{U}$ 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

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

Definition of category, space-based

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

- a **space** $\mathscr{C}_0 : \mathscr{U}$ of objects
- for any $a, b : \mathscr{C}_0$, a **space** $\mathscr{C}(a,b) : \mathscr{U}$ 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

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

- $\mathscr{C}(a,b)$ should be a discrete space (a set in the sense of UF)
- $a = b \to \mathrm{iso}(a,b)$ should be an equivalence

About space-based mathematics

- Univalent foundations is not the origin of the idea of space-based mathematics
- In 2001, Charles Rezk developed **complete Segal spaces** as models for $(\infty, 1)$-categories
  - Rezk's spaces were themselves built from sets
  - Our space-based notion of category is a truncated version of Rezk's complete Segal spaces
- H-spaces: space-based groups
- The book "Symmetry" develops space-based group theory `https://github.com/UniMath/SymmetryBook`

Categories in univalent foundations

An equivalence principle for "complete" categories

$$(A \simeq B) \quad \to \quad \forall P, (P(A) \leftrightarrow P(B))$$

Summary

- \+ All properties P definable in UF are invariant
- \+ Constructions also transfer along equivalence
- \+ "Free completion" operation to build complete categories
- \+ Universal objects (limits) unique up to $=$
- \+ Ess. surj. and f. f. functor admits a quasi-inverse without AC
- − Completeness condition can be difficult to prove

Outline

The need for technology

- Showing that

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

  is an equivalence is difficult when a and b are complicated, "large" structures (consisting of many spaces and operations).
- Stratifying the category of these objects into layers, and proving each layer complete, helps modularize the proof

Example (Category of groups)

The category of groups can be constructed as follows:

- Category of sets
+ "displayed" category of binary operations
+ "displayed" category of unary operations
+ "displayed" category of nullary operations
+ "displayed" category of laws

Completeness of complicated categories

Definition (Displayed category over a category $\mathscr{C}$)

...

Definition (Total category of a displayed category D over $\mathscr{C}$)

Given a category $\mathscr{C}$ and a displayed category $\mathscr{D}$ over it, the **total category** has, as spaces of objects and morphisms, suitable pairs (c, d) of objects and morphisms in $\mathscr{C}$ and $\mathscr{D}$.

- Building blocks can be re-used (e.g., displayed category of binary operations over category of sets)
- Does not rely on meta-theoretic formalization technology
- Generalizes to higher categories

Category theory in UniMath

- UniMath is intended to be a library of all mathematics
- Currently has 345457 loc, of which 247490 loc (71.6 %) on (bi)category theory
- Development of the theory is driven by applications, much of it in the area of meta-theory of programming languages

Challenges

1. Performance bottleneck: type-checking of constructions using large structures is slow
2. $(\infty, 1)$-categories: writing down coherence conditions seems impossible without extending the theory

Summary

1. Building mathematical objects from spaces instead of sets has some advantages, specifically when working with (higher) categories
2. Univalent Foundations and proof assistants built on UF have spaces as primitive objects
3. There is some cost in building things from spaces, but that cost can be managed

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*
- Ahrens, Lumsdaine, *Displayed categories*
- Ahrens, Frumin, Maggesi, Van der Weide, *Bicategories in univalent foundations*