Higher Observational Type Theory

An autonomous foundation for univalent mathematics

Michael Shulman
University of San Diego

j.w.w. Thorsten Altenkirch and Ambrus Kaposi

September 23, 2022
OCIE seminar in History and Philosophy of Logic and Mathematics
Chapman University

Outline

Univalent foundations

Homotopy type theory and univalent foundations (HoTT/UF) is a new approach to the foundations of mathematics.

- Originated 10–15 years ago (Awodey, Voevodsky, and others).
- Its basic objects are homotopy types, a.k.a. ∞-groupoids, rather than sets.

Advantages include:

- Isomorphism-invariant (i.e. representation-independent).
- Well-adapted to computer formalization.
- Has a plethora of useful nonstandard models (higher topoi).
- Supports an entirely new subject (synthetic homotopy theory).

Isomorphism invariance

A basic principle of group theory

If $G \cong H$ are isomorphic groups, and G has some property (cyclic, abelian, solvable, simple, ...), then so does H.

A counterexample to this principle

$\mathbb{Z} \cong 2\mathbb{Z}$ as groups, but $1 \in \mathbb{Z}$ and $1 \notin 2\mathbb{Z}$.

Idea of univalent mathematics

Instead of sets, we build structures out of (homotopy) types.

- No absolute membership; an element has *only one* type.
- In addition to elements, a type has identifications.
  - In familiar types like $\mathbb{N}, \mathbb{Z}, \mathbb{R}$, identifications are just equalities.
  - In the type $\mathbb{G}\text{roup}$ *of groups*, identifications are *isomorphisms*.
- Absolutely anything can be transported across an identification.

Formal systems for univalent mathematics

This general idea can be formalized in multiple ways.

- "Book" Homotopy Type Theory (after the eponymous 2013 book)
- Cubical Type Theories (CCHM 2015, ABCFHL 2021, . . .)
- Higher Observational Type Theory (HOTT)
  (work in progress by Altenkirch-Kaposi-Shulman, 2022)

Each has advantages and disadvantages. We claim an advantage for HOTT that it is easier to understand and justify from first principles.

Today's Goal

Explain and justify HOTT from basic principles, without presupposing either set theory or homotopy theory.

Outline

Constructions and proofs

Observing mathematical practice, we see two activities interspersed:

1. Definitions, a.k.a. constructions. For example:
    - p is "prime" if its only factors are 1 and p.
    - Voevodsky's construction of the derived category of motives.
2. Proofs. For example:
    - Euclid's proof of the infinitude of primes.
    - Voevodsky's proof of the Milnor conjecture using motives.

However, the boundary is not always clearly drawn:

- A construction can depend on a proof: defining $\exp(x) = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$ requires a proof that this series converges.
- A proof may involve intermediate constructions: $M = p_1 p_2 \cdots p_n + 1$ in the infinitude of primes.
- A theorem may claim some object *exists*, but later we may use the *specific object* constructed in the proof.

Propositions-as-types, proofs-as-terms, Curry–Howard, Martin-Löf

In dependent type theory (including HOTT), we unify proofs with constructions. That is, a proof is a particular construction.

Constructions have types

When a proof is called for, it can't be just any proof: it has to be a proof of the desired theorem. Similarly, when a definition of a real number is called for, I can't construct a complex number instead.

Zeroth principle of HOTT (or any dependent type theory)

Each construction/proof has a type. Doing mathematics consists of

1. Specifying a type, and then
2. Giving a construction belonging to it (called a term).

When a term t has type A, we write $t : A$.

Some types are like sets: $\mathbb{N}$, $\mathbb{R}$, $\mathbb{C}$. We call their terms elements.

Other types are like theorems: "There are infinitely many primes", "$\neg\exists xyzn.\ n > 2 \wedge x^n + y^n = z^n$". We call their terms proofs.

Types of hypotheticals

Propositions-as-types unifies all kinds of hypothetical construction:

1. Function definitions, $f(x) = x^2$ (x^2 hypothetical on x)
2. Proofs of $P \Rightarrow Q$ (proving Q hypothetical on truth of P)
3. Proofs of $\forall x.P(x)$ (proving $P(x)$ hypothetical on x)

The type of hypothetical constructions is called a Π-type. Its terms are λ-abstractions.

$$(\lambda x.t[x]) : \prod_{x:A} B[x].$$

Here $t[x]$ constructs an element of $B[x]$, hypothetical on $x : A$.

If B is constant, we write $\prod_{x:A} B$ as $A \to B$.

Types of tuples

Propositions-as-types also unifies all kinds of tuples:

1. Ordered tuples $(2, 4, 7)$ in geometric n-space
2. Bundling sets with structure, e.g. a group (G, m, e, i).
3. Proofs of $P \wedge Q$ (proof of P tupled with proof of Q)
4. Proofs* of $\exists x.P(x)$ (an element c tupled with a proof of $P(c)$)

The type of pairs is called a Σ-type:

$$(a, b) : \sum_{x:A} B[x]$$

Here $a : A$ and $b : B[a]$. If B is constant, write $\sum_{x:A} B$ as $A \times B$.

Higher-ary tuples belong to iterated Σ-types:

$$(a, b, c) \overset{\text{def}}{=} (a, (b, c)) : \sum_{x:A} \sum_{y:B[x]} C[x, y].$$

Types of types

"Specifying a type" is itself a mathematical activity, i.e. a construction. Thus, types are also terms; their type is a universe $\mathscr{U}$.

("$\mathscr{U} : \mathscr{U}$" would have the same problems as a set of all sets, so we actually have a hierarchy of universes $\mathscr{U}_0 : \mathscr{U}_1 : \mathscr{U}_2 : \cdots$.)

(There are many other kinds of types — quotients, disjoint unions, the natural numbers, well-founded trees — but these are enough for now.)

Outline

Equality

One of the most important kinds of proof in mathematics is a proof of equality. Since proofs have types, we need a type of equalities.

First principle of HOTT

For any type A and terms $a, b : A$, there is an identity type

$$a =_A b \quad : \mathscr{U}\,.$$

Its terms are called identifications of a with b.

For specific types A, we *define* the meaning of $a =_A b$ appropriately for terms of that type (examples momentarily).

Reflexivity

The most fundamental property of equality is that everything is equal to itself.

Second principle of HOTT

For any term $a : A$, there is a specified reflexivity term

$$\mathsf{refl}_a : a =_A a.$$

For specific terms a, we define the meaning of refl_a appropriately (examples momentarily).

We also expect equality to be symmetric, transitive, substitutive, etc. These will be deduced later.

Identifications of tuples

Example

An identification of tuples is a tuple of identifications:

$$\Big((a_0, b_0) =_{A \times B} (a_1, b_1)\Big) \overset{\text{def}}{=} \Big((a_0 =_A a_1) \times (b_0 =_B b_1)\Big)$$

$$\mathsf{refl}_{(a,b)} \overset{\text{def}}{=} (\mathsf{refl}_a, \mathsf{refl}_b)$$

In other words, two ordered pairs are equal just when their components are equal pairwise.

(We postpone the general case with B nonconstant.)

Identifications of functions

Example

You might expect equality of functions to be pointwise:

$$¿ \quad (f =_{A\to B} g) \overset{\text{def}}{=} \prod_{x:A} \Big(f(x) =_B g(x)\Big) \quad ?$$

However, we actually make it more general: two functions are equal if they map equal points to equal points.

$$(f =_{A\to B} g) \overset{\text{def}}{=} \prod_{x:A} \prod_{y:A} \Big((x =_A y) \to \big(f(x) =_B g(y)\big)\Big)$$

This way we kill two birds with one stone:

- It *includes* pointwise equality: apply to x, x, and refl_x.
- Reflexivity proves that functions respect equality:

$$\mathsf{refl}_f : \prod_{x:A} \prod_{y:A} \Big((x =_A y) \to \big(f(x) =_B f(y)\big)\Big).$$

Homotopy types

Since $a =_A b$ is itself a type, it has its own identifications.
Thus, a general type A consists of:

- Terms, $a : A$.
- Identifications, $p : a =_A b$.
- Identifications of identifications, $r : p =_{(a =_A b)} q$.
- And so on. . .

Often, only a small amount of this structure matters.

- In $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, etc., each $a =_A b$ has at most one term, and higher identity types are trivial. These are called 0-types.
- In types whose elements are *structures*, such as $\mathbb{G}$roup, $\mathbb{V}$ect, $\mathbb{M}$fd, each $a =_A b$ is a 0-type. These are called 1-types.
- The *type of 1-types* is a 2-type, etc.
- The fundamental homotopy type of a topological space (points, paths, homotopies, etc.) need not be an n-type for any $n < \infty$.

The rest of a homotopy type

Based on our intuition and experience of equality and isomorphism, we expect to have properties like:

- Each identity type is reflexive, symmetric, and transitive.

$$* : (x =_A y) \times (y =_A z) \to (x =_A z).$$

For higher types these are operations containing data: *composition* of isomorphisms, *concatenation* of paths, etc., e.g.

$$\circ : (G \cong H) \times (H \cong K) \to (G \cong K)$$

- These operations must satisfy axioms, like associativity.

$$p * (q * r) =_{(x =_A w)} (p * q) * r.$$

- For higher types, these axioms are themselves operations, and must satisfy their own axioms; etc...

(Un)defining homotopy types

There exist definitions of "homotopy type" in terms of sets that explicitly specify all this complicated structure.

In HOTT, types are undefined objects, like "sets" in ZFC.
We then have a few simple rules (a.k.a. principles, axioms) that

1. Are intuitively justifiable.
2. Generate all this complexity emergently.
   (Compare the emergent complexity of the von Neumann hierarchy.)
3. Allow us to construct all the desired examples.

Specifically:

- The first principle, "All types have identity types", automatically generates infinite towers of identifications.
- The remaining principles similarly generate the operations.

Outline

Observational univalence

Question

Recall that types are terms in a universe $\mathscr{U}$. What is $A =_{\mathscr{U}} B$?

You might expect to see

$$¿ \quad (A =_{\mathscr{U}} B) \overset{\text{def}}{=} \cdots \quad ?$$

the way we did for Σ-types and Π-types.

But recall, we *avoid* complex explicit definitions of types, instead giving rules for how types behave and how to construct them.

Similarly, we avoid an explicit definition of identifications of types, instead giving rules for how they behave and how to construct them.

Identifications of types

How should an identification of types behave?

- In general, nontrivial identifications connect distinct representations of, or names for, the same object.
  - $3 \cdot 3$ and $2^3 + 1$ are both representations of 9, so
    $$3 \cdot 3 =_{\mathbb{N}} 2^3 + 1.$$
  - "The morning star" and "the evening star" are both names for Venus, so
    $$\text{The morning star} =_{\text{Planet}} \text{The evening star.}$$
- Thus, an identification in $\mathscr{U}$ should connect two representations of the same type.

Example: rational numbers

- $\mathbb{Q}_{\mathsf{f}} \overset{\text{def}}{=}$ integer fractions $\frac{1}{2}, -\frac{4}{3}, \frac{3}{7}, \ldots$
- $\mathbb{Q}_{\mathsf{d}} \overset{\text{def}}{=}$ finite or repeating decimals 0.5, $-1.\overline{3}$, $0.\overline{428571}, \ldots$

These are two representations of the rational numbers, so we should have an identification

$$e : \mathbb{Q}_{\mathsf{f}} =_{\mathscr{U}} \mathbb{Q}_{\mathsf{d}}$$

This identification is determined by which elements of $\mathbb{Q}_{\mathsf{f}}$ and $\mathbb{Q}_{\mathsf{d}}$ correspond to each other:

$$\frac{1}{2} \sim 0.5 \qquad -\frac{4}{3} \sim -1.\overline{3} \qquad \frac{3}{7} \sim 0.\overline{428571}$$

Definition

A correspondence between A and B is a function $R : A \times B \to \mathscr{U}$. (We can think of $R(a,b)$ as theorem-like, e.g. $a \sim b$ above.)

Bitotal correspondences

Not every correspondence is an identification of types. We need to know, *at least*, that (e.g.) every fraction has *some* repeating decimal, and every repeating decimal can be written as *some* fraction.

Definition

A correspondence $R : A \times B \to \mathscr{U}$ is bitotal if

- For every $a : A$, there is a $b : B$ and an $r : R(a, b)$; and
- For every $b : B$, there is an $a : A$ and an $s : R(a, b)$.

In symbols, we must have a term of

$$\mathrm{isBitotal}(R) \overset{\text{def}}{=} \Big(\textstyle\prod_{a:A} \sum_{b:B} R(a, b)\Big) \times \Big(\textstyle\prod_{b:B} \sum_{a:A} R(a, b)\Big).$$

Identifications of types

Third principle of HOTT

Every identification of types $e : A =_{\mathscr{U}} B$ is equipped with a bitotal correspondence

$$\llbracket e \rrbracket : A \times B \to \mathscr{U}.$$

Of course, any type $A : \mathscr{U}$ has a reflexivity identification

$$\mathsf{refl}_A : A =_{\mathscr{U}} A.$$

There is an obvious choice for its underlying correspondence:

Fourth principle of HOTT

For $A : \mathscr{U}$, the correspondence of refl_A is its identity types:

$$\llbracket \mathsf{refl}_A \rrbracket(a, b) \overset{\text{def}}{=} (a =_A b).$$

Substitution

This seems like very little, but it's almost all we need!
First, note for $B : A \to \mathscr{U}$ we have

$$\mathsf{refl}_B : \prod_{a_0 : A} \prod_{a_1 : A} \left((a_0 =_A a_1) \to (B(a_0) =_{\mathscr{U}} B(a_1)) \right).$$

Theorem

Identifications are substitutive: given $B : A \to \mathscr{U}$, with $b_0 : B(a_0)$ and $a_2 : a_0 =_A a_1$, we have a term in $B(a_1)$.

Proof.

Since $\mathsf{refl}_B(a_0, a_1, a_2) : B(a_0) =_{\mathscr{U}} B(a_1)$, there is a bitotal correspondence $[\![\mathsf{refl}_B(a_0, a_1, a_2)]\!]$ between $B(a_0)$ and $B(a_1)$. Thus, any $b_0 : B(a_0)$ corresponds to some term of $B(a_1)$. $\square$

Dependent identifications

Definition

For $b_0 : B(a_0)$ and $b_1 : B(a_1)$, terms of $[\![\mathsf{refl}_B(a_0, a_1, a_2)]\!](b_0, b_1)$ are dependent identifications of b_0 with b_1 "along" a_2:

$$\left(b_0 =^{a_2}_B b_1\right) \overset{\text{def}}{=} [\![\mathsf{refl}_B(a_0, a_1, a_2)]\!](b_0, b_1).$$

We use these to define identity types of general Σ- and Π-types:

$$\Big((a_0, b_0) =_{\sum_{x:A} B(x)} (a_1, b_1)\Big) \overset{\text{def}}{=} \sum_{a_2 : a_0 =_A a_1} \left(b_0 =^{a_2}_B b_1\right)$$

$$\Big(f =_{\prod_{x:A} B(x)} g\Big) \overset{\text{def}}{=} \prod_{a_0:A} \prod_{a_1:A} \prod_{a_2 : a_0 =_A a_1} \left(f(a_0) =^{a_2}_B g(a_1)\right).$$

Specialized dependent identifications

The definitions of refl by term imply that

- $\mathsf{refl}_f(a, a, \mathsf{refl}_a) \overset{\text{def}}{=} \mathsf{refl}_{f(a)}$.
- $\mathsf{refl}_{\lambda x.b}(a_0, a_1, a_2) \overset{\text{def}}{=} \mathsf{refl}_b$ for b independent of x.
- $\mathsf{refl}_{\lambda x.x}(a_0, a_1, a_2) \overset{\text{def}}{=} a_2$.

Combined with the fourth principle, we get:

- $(b_0 =_B^{\mathsf{refl}_a} b_1) \overset{\text{def}}{=} (b_0 =_{B(a)} b_1)$.
- $(b_0 =_{\lambda x.B}^{a_2} b_1) \overset{\text{def}}{=} (b_0 =_B b_1)$ for $B : \mathscr{U}$ a constant type.
- $(b_0 =_{\lambda X.X}^{e} b_1) \overset{\text{def}}{=} [\![e]\!](b_0, b_1)$ for $e : B_0 =_{\mathscr{U}} B_1$.

In particular, the identity types of $\sum_{x:A} B(x)$ and $\prod_{x:A} B(x)$ specialize to those of $A \times B$ and $A \to B$.

Transitivity and symmetry

Theorem

Identification is transitive and symmetric.

Proof.

Consider the family of identity types:

$$\mathsf{Id}_A : A \times A \to \mathscr{U}$$
$$\mathsf{Id}_A(x, y) \overset{\text{def}}{=} (x =_A y).$$

Substitution in Id_A along $(r_0, r_1) : (a_0, a_1) =_{A \times A} (b_0, b_1)$ says that any $s : a_0 =_A a_1$ yields a term of $b_0 =_A b_1$.

- Taking $b_0 \overset{\text{def}}{=} a_0$ and $r_0 \overset{\text{def}}{=} \mathsf{refl}_{a_0}$ yields transitivity.
- Taking $a_1 \overset{\text{def}}{=} b_1 \overset{\text{def}}{=} a_0$ and $r_1 \overset{\text{def}}{=} s \overset{\text{def}}{=} \mathsf{refl}_{a_0}$ yields symmetry. □

We get all the higher structure of a homotopy type in a similar way.

Identifying identity types

More generally, consider $\mathsf{Id} : \left(\sum_{X:\mathscr{U}}(X \times X)\right) \to \mathscr{U}$ defined by

$$\mathsf{Id}(X, x, y) \overset{\text{def}}{=} (x =_X y).$$

For (A, a_0, a_1) and (B, b_0, b_1) in $\sum_{X:\mathscr{U}}(X \times X)$, we have

$$\Big((A, a_0, a_1) =_{\sum_{X:\mathscr{U}}(X\times X)} (B, b_0, b_1)\Big) \overset{\text{def}}{=} \\ \sum_{e:A=_{\mathscr{U}}B} \left([\![e]\!](a_0, b_0) \times [\![e]\!](a_1, b_1)\right).$$

Thus if we have $r_0 : [\![e]\!](a_0, b_0)$ and $r_1 : [\![e]\!](a_1, b_1)$, for some identification of types $e : A =_{\mathscr{U}} B$, then we obtain

$$\mathsf{refl}_{\mathsf{Id}}((A, a_0, a_1), (B, b_0, b_1), (e, r_0, r_1)) : (a_0 =_A a_1) =_{\mathscr{U}} (b_0 =_B b_1).$$

One-to-one correspondences

In particular, given $r_0 : [\![e]\!](a_0, b_0)$ and $r_1 : [\![e]\!](a_1, b_1)$, we have $a_0 =_A a_1$ if and only if $b_0 =_B b_1$. This is essentially the standard notion of one-to-one correspondence.

Example

- We have $\frac{1}{2} \sim 0.5$ and $\frac{1}{2} \sim 0.4\overline{9}$, so $\mathrm{refl}_{\frac{1}{2}}$ yields $0.5 =_{\mathbb{Q}_\mathrm{d}} 0.4\overline{9}$.
- We have $\frac{1}{3} \sim 0.\overline{3}$ and $\frac{3}{9} \sim 0.\overline{3}$, so $\mathrm{refl}_{0.\overline{3}}$ yields $\frac{1}{3} =_{\mathbb{Q}_\mathrm{f}} \frac{3}{9}$.

If A and B are 0-types, like $\mathbb{Q}_\mathrm{f}$ and $\mathbb{Q}_\mathrm{d}$, nothing else is needed. But for general types, we need to know the whole identification

$$(a_0 =_A a_1) =_{\mathscr{U}} (b_0 =_B b_1).$$

From bijections to identifications

- Just as Principles 0, 1, and 2 tell us how types behave, Principles 3 and 4 tell us how identifications of types behave.
- Just as Π, Σ, etc. construct particular types, we can construct particular identifications of types:

Theorem

Given $f : A \to B$ and $g : B \to A$ such that $g \circ f =_{A \to A} \mathrm{id}_A$ and $f \circ g =_{B \to B} \mathrm{id}_B$ (a bijection), we have $\langle f \rangle : A =_{\mathscr{U}} B$.

Proof.

Define $[\![\langle f \rangle]\!](a, b) \overset{\text{def}}{=} (f(a) =_B b)$. This is bitotal because:

- Given $a : A$, we have $f(a) : B$ and $\mathrm{refl}_{f(a)} : f(a) =_B b$.
- Given $b : B$, we have $g(b) : A$ and $f(g(b)) =_B b$.

The rest of the data is similar. □

An identification of types "records" much more structure than a bijection, but in practice we usually generate it from one.

Conclusion

Theorem

These basic principles, plus the omitted details, imply that every type has all the expected operations and axioms at all levels.

Thus, although it takes a bit of work, our basic principles generate a complete theory of homotopy types, without presupposing either set theory or homotopy theory.

The resulting system can serve as an isomorphism-invariant foundation for mathematics.