Perry Hart
PLClub
Talk #1
February 1, 2019

Abstract

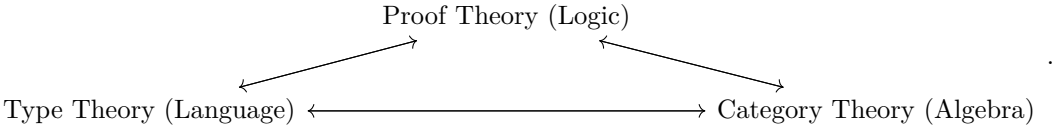
This is a brief, introductory overview of homotopy type theory (HoTT). After covering some preliminary concepts from type theory, we work to state the *univalence axiom*, a strong feature of HoTT as formulated by Voevodsky. In doing this, we develop the notion both of an identity type and of type equivalence by way of homotopy theory (an area of algebraic topology). Afterward, we mention some categorical models of type theory that have been particularly significant to HoTT.

Contents

Background

In its current form, HoTT largely originated from Vladimir Voevodsky, a Fields Medalist who passed away in 2017. He wanted a logical system that could both handle certain higher-dimensional algebraic objects and facilitate formalization better than set theory could. In 2005 and 2006, the first homotopical models of intensional (dependent) type theory were constructed by Steve Awodey, Michael Warren, and Voevodsky. In 2012-13, the IAS at Princeton held *A Special Year on Univalent Foundations of Mathematics*, from which the standard reference *Homotopy Type Theory: Univalent Foundations of Mathematics* was produced. This includes Voevodsky's univalence axiom in its formulation of type theory.

One may think of HoTT as a topological (or, more generally, homotopical) version of what Robert Harper calls "computational trinitarianism":



1 Elements of type theory

We are given a formal language $\mathcal{L}$ consisting of certain terms, say, a variant of the untyped lambda calculus. We can enrich $\mathcal{L}$ with additional primitive objects called *types*. If a is a term and A a type, then we write $a : A$ to express the *judgment* (distinct from a proposition) that a *inhabits* or *has type* A. We declare that types themselves inhabit types known as *universes*, which are arranged in a cumulative hierarchy.

$$\mathbf{Type}_0 : \mathbf{Type}_1 : \mathbf{Type}_2 : \cdots .$$

If instead we declared a single universe **Type** that every type inhabits, then we could encode Russel's paradox into our type theory. Our hierarchy avoids such a problem by tracking which level a newly formed type inhabits. For convenience, however, we usually avoid writing the level explicitly and use the term **Type** instead.

Example 1.0.1. The following (among others) will be base types of our language.

1. The *empty type* $\bot$.
2. The *one point type* $*$.

We write $a \equiv b : A$ to express that the terms a and b are *definitionally equal with respect to the type* A. We declare that $\equiv$ is an equivalence relation with respect to a fixed type A. We also declare that if $A \equiv B : \mathbf{Type}$ and $a : A$, then $a : B$.

Definition 1.0.2. Let A be any type and $B : A \to \mathbf{Type}$ be any family of types (e.g., $\lambda n.T^n : \mathbb{N} \to \mathbf{Type}$ when $A \equiv \mathbb{N} : \mathbf{Type}$).

1. We form the *dependent product type* $\prod_{x:A} B(x)$ according to the following four rules.
   (a) $\prod$-Introduction: If $b : B(x)$ for any $x : A$, then $\lambda x.b : \prod_{x:A} B(x)$.
   (b) $\prod$-Elimination: If $f : \prod_{x:A} B(x)$ and $a : A$, then $f(a) : B(a)$.
   (c) $\prod$-Computation: If $b : B(x)$ for any $x : A$ and $a : A$, then $(\lambda x.b)(a) \equiv b[x \coloneqq a] : B(a)$.
   (d) $\prod$-Uniqueness: If $f : \prod_{x:A} B(x)$, then $f \equiv \lambda x.f(x) : \prod_{x:A} B(x)$.

   Note that the ordinary function type $A \to B$ is a special case of the dependent product type. Naively, a dependent product type is comparable to a set-theoretic choice function.
2. Now, add the term $\mathrm{ind}_{\sum_{x:A} B(x)}(t_1, t_2, t_3)$ to $\mathcal{L}$ where t_1, t_2, and t_3 are any given terms of $\mathcal{L}$. We form the *dependent sum type* $\sum_{x:A} B(x)$ according to the following three rules.

(a) $\sum$-Introduction: If $a : A$ and $b : B(a)$, then $(a, b) : \sum_{x:A} B(x)$.

(b) $\sum$-Elimination: Given any family of types $C : (\sum_{x:A} B(x)) \to \mathbf{Type}$, if $g : C(x, y)$ for any $x : A$ and $y : B(x)$ and $p : \sum_{x:A} B(x)$, then $\mathrm{ind}_{\sum_{x:A} B(x)}(C, g, p) : C(p)$.

(c) $\sum$-Computation: Given any family of types $C : (\sum_{x:A} B(x)) \to \mathbf{Type}$, if $g : C(x, y)$ for any $x : A$ and $y : B(x)$, $a : A$, and $b : B(a)$, then $\mathrm{ind}_{\sum_{x:A} B(x)}(C, g, (a, b)) \equiv g(a, b) : C(a, b)$.

Note that the ordinary product type $A \times B$ is a special case of the dependent sum type. Naively, a dependent sum type is comparable to a set-theoretic disjoint union (or, more generally, as a coproduct).

Definition 1.0.3.

1. Define the *left projection function* pr_1 by the judgments
$$\mathrm{pr}_1 : \left(\sum_{x:A} B(x)\right) \to A \qquad \mathrm{pr}_1(a, b) \equiv a : A.$$
2. Define the *right projection function* pr_2 by the judgements
$$\mathrm{pr}_2 : \prod_{p:\sum_{x:A} B(x)} B(\mathrm{pr}_1(p)) \qquad \mathrm{pr}_2(a, b) \equiv b : B(a).$$

Given $p : \sum_{x:A} B(x)$, we have that $p \equiv (\mathrm{pr}_1(p), \mathrm{pr}_2(p)) : \sum_{x:A} B(x)$. This is known as the uniqueness principle for dependent sum types.

Remark 1.0.4. Under the Curry-Howard isomorphism, we get the following propositional interpretation of dependent type theory.

$$\text{False} \longleftrightarrow \bot$$

$$\text{True} \longleftrightarrow *$$

$$P \wedge Q \longleftrightarrow P \times Q$$

$$P \vee Q \longleftrightarrow P + Q \quad .$$

$$P \implies Q \longleftrightarrow P \to Q$$

$$\forall x.P(x) \longleftrightarrow \textstyle\prod_{x:A} P(x)$$

$$\exists x.P(x) \longleftrightarrow \textstyle\sum_{x:A} P(x)$$

2 Identity types

Definition 2.0.1. Let A be a type. The *identity type* $\mathrm{Id}_A : A \to A \to \mathbf{Type}$ is given inductively by the single constructor
$$\mathsf{refl} : \prod_{a:A} \mathrm{Id}_A(a, a).$$

We interpret A to be a topological space and each inhabitant of $\mathrm{Id}_A(x, y)$ to be a path from the point x to the point y. The term refl_a is thought of as the constant path at the point a. Let $x \rightsquigarrow_A y$ denote the type $\mathrm{Id}_A(x, y)$.

Note 2.0.2. The type $x \rightsquigarrow_A y$ can be thought of as the path space of A, which consists of the set of paths in A equipped with the compact-open topology.

Remark 2.0.3. Under Curry-Howard, each path from x to y corresponds to a proof of the proposition $x = y : A$. Thus, HoTT is an intensional type theory in that it both distinguishes between definitional and propositional equality and allows an identity type to be inhabited by more than one term.

Remark 2.0.4 (Path induction). Given a dependent type $D : \prod \{x, y : A\}, \mathrm{Id}_A(x, y) \to \mathbf{Type}$, suppose that there is some $d : \prod_{a:A} D(a, a, \mathsf{refl}_a)$. Then the induction principle for identity types states that there exists a *section*

$$J(D, d) : \prod_{\substack{\{x,y:A\} \\ p:\mathrm{Id}_A(x,y)}} D(x, y, p)$$

of D such that $J(D, d, \mathsf{refl}_a) = d(a)$ for each $a : A$.

Lemma 2.0.5. *Let* $A : \mathbf{Type}$ *and* $x, y : A$. *Then there is some function* $\mathsf{sym} : (x \rightsquigarrow y) \to (y \rightsquigarrow x)$ *such that* $\mathsf{sym}(\mathsf{refl}_x) \equiv \mathsf{refl}_x$ *for each* $x : A$. *Let* $p^{-1} \coloneqq \mathsf{sym}(p)$.

Proof. Define $D : \prod(x, y : A), (x \rightsquigarrow y) \to \mathbf{Type}$ by the judgment

$$D(x, y, p) \equiv y \rightsquigarrow x.$$

By our construction of Id, we have the function term

$$d \coloneqq \lambda x.\, \mathsf{refl}_x : \prod_{x:A} D(x, x, \mathsf{refl}_x).$$

By induction, we obtain an extension J of d such that $J(D, d, x, y, p) : y \rightsquigarrow x$ for every $p : x \rightsquigarrow y$. Letting $\mathsf{inv}(p) \coloneqq J(D, d, x, y, p)$ completes the proof. □

Lemma 2.0.6. *Let* $A : \mathbf{Type}$ *and* $x, y, z : A$. *Then there is some function* $\mathsf{trans} : (x \rightsquigarrow y) \to (y \rightsquigarrow z) \to (x \rightsquigarrow z)$ *such that* $\mathsf{trans}(q, \mathsf{refl}_z) \equiv q$ *for any* $q : y \rightsquigarrow z$. *Let* $p * q \coloneqq \mathsf{trans}(p, q)$.

Proof. Define the dependent type D over Id_A by

$$D(x, y, q) \equiv (y \rightsquigarrow z) \to (x \rightsquigarrow z).$$

We have the function term

$$d \coloneqq \lambda z.\mathrm{idmap}_{y \rightsquigarrow z} : \prod_{z:A} D(z, z, \mathsf{refl}_z)$$

where $\mathrm{idmap}_{y \rightsquigarrow z} \coloneqq \lambda q.q$. By induction, we get a suitable section J of D such that $J(D, d, p) : (y \rightsquigarrow z) \to (x \rightsquigarrow z)$ for each $p : x \rightsquigarrow y$. Finally, let $\mathsf{trans}(p, -) \coloneqq (D, d, p)(-)$. □

Remark 2.0.7. Also, we can use the induction principle to show that A is a groupoid whose structure is exactly similar to the groupoid structure of a topological space X, which we recall now.

1. Let $\gamma : I \to X$ in X be a path in X. Then $\bar{\gamma} \coloneqq \gamma(1 - t)$ is a path from $\gamma(1)$ to $\gamma(0)$, called the *inverse of* γ. This fact corresponds to the function sym.
2. Let $\phi, \psi : I \to X$ be paths in the topological space X such that $\phi(1) = \psi(0)$. Then $\phi * \psi : I \to X$ defined by $t \mapsto \begin{cases} \phi(2t) & 0 \le t \le \frac{1}{2} \\ \psi(2t - 1) & \frac{1}{2} \le t \le 1 \end{cases}$ is a path from $\phi(0)$ to $\psi(1)$. This fact corresponds to the function trans.
3. Define the *fundamental groupoid of a space* X as the category $\Pi_1(X)$ with $\mathrm{ob}(\Pi_1(X)) = X$ and $\mathrm{Hom}_{\Pi_1(X)}(x, y) = \{[\gamma]_{\simeq_p} \mid \gamma \text{ is a path from } x \text{ to } y\}$. The composition of morphisms is given by the operation $*$.

It is straightforward but tedious to verify that Π_1 satisfies the definition of a groupoid with $[\bar{\gamma}] = [\gamma]^{-1}$.

Our next result shows that if two terms of type A are propositionally equal and P is a property of inhabitants of A, then P is true of one if and only if it is true of the other.

Lemma 2.0.8 (Transportion). *Let P be a dependent type over A. Suppose that $p : x \rightsquigarrow_A y$. Then there exists a function $\mathsf{transport}(p) : P(x) \to P(y)$ such that $\mathsf{transport}(\mathsf{refl}_x)(u) \equiv u$ for any $u : P(x)$. Let $p \cdot u \coloneqq \mathsf{transport}(p)(u)$.*

Proof. Define $D : \prod(x, y : A), (x \rightsquigarrow y) \to \mathbf{Type}$ by $D(x, y, p) \equiv P(x) \to P(y)$. We have the function $d \coloneqq \lambda x.\mathrm{idmap}_{P(x)} : \prod_{x:A} D(x, x, \mathsf{refl}_x)$. By induction, we get a section J of D that extends d and produces terms $J(D, d, p) : P(x) \to P(y)$ for each $p : x \rightsquigarrow y$. Finally, define $\mathsf{transport}(p) \equiv J(D, d, p)$. □

Lemma 2.0.9. *Let $f : \prod_{x:A} P(x)$ and $p : x \rightsquigarrow_A y$. Then there is some path $f(p) : p \cdot f(x) \rightsquigarrow_{P(y)} f(y)$.*

Proof. Define $D : \prod(x, y : A), (x \rightsquigarrow y) \to \mathbf{Type}$ by $D(x, y, p) \equiv p \cdot f(x) \rightsquigarrow f(y)$. Then

$$D(x, y, \mathsf{refl}_x) \equiv \mathsf{refl}_x \cdot f(x) \rightsquigarrow f(x) \equiv f(x) \rightsquigarrow f(x).$$

We thus obtain the term

$$d \coloneqq \lambda x.\, \mathsf{refl}_{f(x)} : \prod_{x:A} D(x, x, \mathsf{refl}_x).$$

Applying induction finishes the proof. □

Corollary 2.0.10. *Any non-dependent function $f : A \to B$ is functorial, i.e., preserves paths, and therefore is continuous in a certain sense.*

Proof. Simply recall that any non-dependent function is a special case of a dependent one. □

Lemma 2.0.11 (Path lifting). *Let P be a dependent type over A. Suppose that $p : x \rightsquigarrow_A y$ and that $u : P(x)$. Then there is some path*

$$p_\Sigma(u) : (x, u) \rightsquigarrow_{\sum_{x:A} P(x)} (y, p \cdot u).$$

Proof. Define $D : \prod(x, y : A), (x \rightsquigarrow y) \to \mathbf{Type}$ by

$$D(x, y, p) \equiv \prod(u : P(x)), (x, u) \rightsquigarrow (y, p \cdot u).$$

Since $\mathsf{refl}_x \cdot u \equiv u$ for any $u : P(x)$, we have that $D(x, x, \mathsf{refl}_x) \equiv \prod(u : P(x)), (x, u) \rightsquigarrow (x, u)$. But then

$$d \coloneqq \lambda x \lambda u.\, \mathsf{refl}_{(x,u)} : \prod_{x:A} D(x, x, \mathsf{refl}_x).$$

By induction, we obtain some path $p_\Sigma(u) : (x, u) \rightsquigarrow (y, p \cdot u)$ for each $p : x \rightsquigarrow y$. □

As fiber bundles from topology possess the homotopy lifting property, our last lemma encourages us to interpret the dependent sum $\sum_{x:A} P(x)$ as a fiber bundle over A.

Lemma 2.0.12. *Let $f, g : \prod_{x:A} P(x)$ where P is a dependent type over A. Any $\alpha : f \rightsquigarrow g$ induces paths $\alpha(x) : f(x) \rightsquigarrow_{P(x)} g(x)$ for each $x : A$. Therefore, we get a function*

$$\mathsf{hApply} : \prod\{A : \mathbf{Type}\}\{P : A \to \mathbf{Type}\}(f, g : \prod(x : A), P(x)), (f \rightsquigarrow g) \to \left(\prod(x : A) f(x) \rightsquigarrow g(x)\right).$$

Proof. An easy use of path induction. □

Definition 2.0.13. Let $f, g : \prod_{x:A} P(x)$ where P is a dependent type over A. A *homotopy from f to g* is a term H of type

$$\prod(x : A), f(x) \rightsquigarrow_{P(x)} g(x).$$

We say that f and g are *homotopic* and denote the space of homotopies from f to g by $f \simeq g$.

Definition 2.0.14. Let $u, v, x, y : A$. We say that the diagram

$$\begin{array}{ccc} v & \overset{q}{\rightsquigarrow} & y \\ {\scriptstyle p}\uparrow & & \uparrow{\scriptstyle s} \\ x & \underset{r}{\rightsquigarrow} & u \end{array}$$

commutes if there is some term $h : p * q \rightsquigarrow r * s$, called a *witness of the commutativity.*

Lemma 2.0.15. *Let $f, g : A \to B$. Suppose that $H : f \simeq g$. The for each path $p : x \rightsquigarrow_A y$, the square*

$$\begin{array}{ccc} f(y) & \overset{H(y)}{\rightsquigarrow} & g(y) \\ {\scriptstyle f(p)}\uparrow & & \uparrow{\scriptstyle g(p)} \\ f(x) & \underset{H(x)}{\rightsquigarrow} & g(x) \end{array}$$

commutes.

Proof. Use path induction. □

Note 2.0.16. The naive form of functional extensionality implies that if f and g are homotopic, then they are propositionally equal.

Note 2.0.17. The fact that every function preserves paths shows us, intuitively, that our type-theoretic definition of homotopy is as strong as our topological definition. In fact, we will see that any model of HoTT must satisfy a form of functional extensionality, which ensures that every type-theoretic homotopy induces a continuous choice of paths $f(x) \rightsquigarrow g(x)$, as desired.

The following notion corresponds to homotopy equivalence from topology.

Definition 2.0.18. Let $f : A \to B$ be a function. We say that f is a *(homotopy) isomorphism* if there is some $g : B \to A$ such that $f \circ g \simeq \mathrm{idmap}_B$ and $g \circ f \simeq \mathrm{idmap}_A$. Then the space $\mathsf{iso}(A, B)$ of homotopy isomorphisms from A to B is precisely

$$\sum (f : A \to B)(g : B \to A), (\mathrm{idmap}_B \simeq f \circ g) \times (\mathrm{idmap}_A \simeq g \circ f).$$

Proposition 2.0.19.

1. *Any inverse of $f : A \to B$ is unique up to homotopy.*
2. *Any function homotopic to an isomorphism is an isomorphism (just as a continuous map in topology).*

Definition 2.0.20. We say that a type A is *contractible* if

$$\mathsf{isContr}(A) :\equiv \sum_{a:A} \prod_{x:A} x \rightsquigarrow a$$

is inhabited. In this case, we say that the first component of any inhabitant is the *center of contraction.*

Note 2.0.21. To preserve our topological intuition, we interpret this as saying that A is contractible when there is a homotopy from $\lambda x.x$ to $\lambda x.a$ for some $a : A$.

Example 2.0.22. The one point type (denoted by $*$ or unit) is contractible.

Proof. This is easy to show by induction on unit. □

3 Type-theoretic equivalence

We begin this section with a version of the set-theoretic inverse image.

Definition 3.0.1. Let $f : A \to B$ be a function and $b : B$. The *homotopy fiber of* b is the space

$$\mathsf{hFiber}(f, b) :\equiv \sum (a : A), f(a) \rightsquigarrow b.$$

Definition 3.0.2. A function $f : A \to B$ is an *equivalence from* A *to* B if $\mathsf{hFiber}(f, b)$ is contractible for each $b : B$, i.e., if some term has type

$$\mathsf{isEquiv}(f) :\equiv \prod (b : B), \mathsf{isContr}(\mathsf{hFiber}(f, b)).$$

In this case, we say that A and B are *equivalent* and write the space of such equivalences as $A \simeq B$.

Our next notion corresponds to the overcategory.

Definition 3.0.3. Let A be a type with $a : A$. Define the dependent type $\mathsf{Y}(a)$ over A by

$$\mathsf{Y}(a)(x) \equiv x \rightsquigarrow a$$

for each $x : A$.

Lemma 3.0.4. *If* $p : x \rightsquigarrow_A y$, *then* $\underbrace{p \cdot p}_{w.r.t.\ \mathsf{Y}(y)} \rightsquigarrow_{y \rightsquigarrow y} \mathsf{refl}_y$ *is inhabited.*

Proof. Use path induction. □

Example 3.0.5. For any type A, the identity map $A \to A$ is an equivalence.

Proof. We must show that $\mathsf{hFiber}(\mathrm{idmap}_A, a)$ is contractible for each $a : A$. Let $p : x \rightsquigarrow_A a$, so that (x, p) is a canonical inhabitant of $\mathsf{hFiber}(\mathrm{idmap}_A, a)$. By the path lifting lemma, we know that $p_\Sigma(p) : (x, p) \rightsquigarrow (a, p \cdot p)$. Our previous lemma implies that there exists a term of type $p \cdot p \rightsquigarrow_{a \rightsquigarrow a} \mathsf{refl}_a$. Hence $(x, p) \rightsquigarrow (a, \mathsf{refl}_a)$ is inhabited, making (a, refl_a) the center of contraction. □

Corollary 3.0.6. *Let* P *be a dependent type over* A *and* $p : x \rightsquigarrow_A y$. *Then* $\mathsf{transport}(p)$ *is an equivalence from* $P(x)$ *to* $P(y)$.

Our next concept is similar to a unit-counit adjunction in category theory.

Definition 3.0.7. An *adjoint equivalence from* A *to* B is a 5-tuple $(f, g, \eta, \epsilon, \alpha)$ consisting of $f : A \to B$, $g : B \to A$, $\eta : \mathrm{idmap}_A \simeq g \circ f$, $\epsilon f \circ g \simeq \mathrm{idmap}_B$, and witnesses $\alpha(x)$ of the commutativity of the triangle

$$\begin{array}{ccc} f(x) & \overset{f(\eta(x))}{\rightsquigarrow} & f(g(f(x))) \\ & {\scriptstyle \mathsf{refl}_{f(x)}} \searrow & \downarrow {\scriptstyle \epsilon(f(x))} \\ & & f(x) \end{array}$$

for each $x : A$.

Theorem 3.0.8. *A function* $f : A \to B$ *is an equivalence if and only if it is an isomorphism.*

Proof. We just prove the ($\Longleftarrow$) direction, which is harder. We derive it from the following fact.

Proposition 3.0.9. *If* $(f, g, \eta, \epsilon, \alpha)$ *is an adjoint equivalence from* A *to* B, *then* f *is an equivalence.*

Now, suppose that $(f, g, H, K) : \mathsf{iso}(A, B)$. We want to construct an adjoint equivalence $(f, g, \eta, \epsilon, \alpha)$. Define η by

$$\eta(x) \equiv H(x) * g(K(f(x)))^{-1} * g(f(H(x)))^{-1}.$$

Let $\epsilon \equiv K$. It remains to check that $f(\eta(x)) * \epsilon(f(x)) \rightsquigarrow \mathsf{refl}_{f(x)}$. To do this, we apply Lemma 2.0.15 after doing a bit of diagram chasing. □

Note 3.0.10. It is not the case that $\mathsf{iso}(A, B) \simeq (A \simeq B)$.

Corollary 3.0.11. *Any two contractible types are equivalent.*

Definition 3.0.12.

1. The *weak functional extensionality principle (WFE)* is that for any dependent type P over A, the space $\prod_{x:A} P(x)$ is contractible whenever each $P(x)$ is contractible.
2. The *strong functional extensional principle (SFE)* is that there is some term of type

$$\prod\{A : \mathbf{Type}\}\{P : A \to \mathbf{Type}\}(f, g : \prod_{x:A} P(x)), \mathsf{isEquiv}(\mathsf{hApply}(f, g)).$$

Note that any product of contractible spaces is contractible in traditional topology.

Proposition 3.0.13 (Axiom of choice). *Let P be a dependent type over A and $R : \prod(x : A), P(x) \to$ **Type**. Then there is some term of type*

$$\prod_{x:A} \sum_{u:P(x)} R(x, u) \to \sum(s : \prod_{x:A} P(x)) \prod(x : A), R(x, s(x)).$$

Definition 3.0.14. Let P and Q be dependent spaces over A. Suppose that $\tau : \prod(x : A), P(x) \to Q(x)$. Define $\Sigma_A \tau : \sum_{x:A} P(x) \to \sum_{x:A} Q(x)$ by $\lambda w.(\mathrm{pr}_1\, w, \tau(\mathrm{pr}_1\, w)(\mathrm{pr}_2\, w))$.

Theorem 3.0.15 (Voevodsky). *Let P and Q be dependent spaces over A. If a term $\tau : \prod(x : A), P(x) \to Q(x)$ is such that $\Sigma_A \tau$ is an equivalence, then τ is a fiberwise equivalence, i.e., $\tau(x)$ is an equivalence for each $x : A$.*

Along with the axiom of choice and the fact that any two contractible spaces are equivalent, this result turns out to be enough to establish that WFE $\implies$ SFE.

Note 3.0.16. We also assume that $\lambda x.f(x) \equiv f$ where f is a dependent product over A, which is not built into Coq.

Theorem 3.0.17. *WFE entails SFE.*

4 Univalence axiom

Lemma 4.0.1. *For any types A and B and any path $p : A \rightsquigarrow_{\mathbf{Type}} B$, there is some $v(A, B, p) : A \simeq B$.*

Proof. Note that the identity map on A is an equivalence. Hence we may apply path induction. □

Definition 4.0.2. The *univalence axiom* is that $v(A, B) : (A \rightsquigarrow B) \to (A \simeq B)$ is an equivalence. In particular, there is some function $v(A, B)^{-1} : (A \simeq B) \to (A \rightsquigarrow B)$.

Lemma 4.0.3 (Induction on equivalences). *Suppose that*

$$D : \prod(A, B : \mathbf{Type}), (A \simeq B) \to \mathbf{Type}$$

and that there is some

$$d : \prod_{A:\mathbf{Type}} D(A, A, idmap_A).$$

Then there is some section

$$J(D, d) : \prod_{\substack{\{A,B:\mathbf{Type}\} \\ e:A\simeq B}} D(A, B, e)$$

of D.

Proof. Let $A, B : \mathbf{Type}$ and $e : A \simeq B$. There is some path $v(A,B)(v(A,B)^{-1}(e)) \rightsquigarrow_{A \simeq B} e$. Thus, it suffices to show that $D(A, B, v(A,B)(v(A,B)^{-1}(e)))$ is inhabited. It follows that it is enough to find a section of the dependent type D' over $A \rightsquigarrow_{\mathbf{Type}} B$ where $D'(A,B,p) \equiv D(A,B,v(A,B,p))$. But note that $D'(A,A,\mathsf{refl}_A) = D(A,A,\mathrm{idmap}_A)$, which is inhabited by assumption. By path induction, we are done. □

We now have a powerful tool for proving things about function spaces.

Corollary 4.0.4. *If $A \simeq B$, then $(X \to A) \simeq (X \to B)$ for each space X.*

Proposition 4.0.5. *The univalence axiom entails naive non-dependent functional extensionality.*

Corollary 4.0.6 (Voevodsky). *The univalence axiom entails WFE (hence SFE).*

Proof. Suppose that $P : A \to \mathbf{Type}$ and that there is some $K : \prod_{x:A} \mathsf{isContr}(P(x))$. We must show that $\prod_{x:A}, P(x)$ is contractible. Define $U : A \to \mathbf{Type}$ the constant function $\lambda x.\mathsf{unit}$. Since both $P(x)$ and unit are contractible for each $x : A$, we have that $P(x) \simeq \mathsf{unit}$. By univalence, it follows that $P(x) \rightsquigarrow \mathsf{unit}$. Our previous proposition implies that $P \rightsquigarrow_{A \to \mathbf{Type}} U$.

Therefore, there is some term of type

$$\left(\prod_{x:A} P(x)\right) \rightsquigarrow \left(\prod_{x:A}, U(x)\right).$$

This shows that it suffices to show that $\prod_{x:A}, U(x)$ is contractible. But $\prod_{x:A}, U(x) \equiv A \to \mathsf{unit}$, and any term of this is homotopic to a constant function. Another use of our last proposition proves that $A \to \mathsf{unit}$ is contractible. □

Finally, by assuming the univalence axiom, we prove a fundamental result of category theory in our type theory.

Lemma 4.0.7 (Yoneda). *Let P and Q be dependent types over A (viewed as type-valued presheaves on A). Let $\mathrm{Hom}(P,A) :\equiv \prod(x : A), P(x) \to Q(x)$.*

Assume that WFE holds. Then for any dependent type P over A and any $a : A$, there exists a natural equivalence

$$\alpha_{P,a} : \mathrm{Hom}(\mathsf{Y}(a), P) \simeq P(a).$$

The term $\alpha_{P,a}$ is natural in the sense that for any $p : a \rightsquigarrow a'$ and $\sigma : \mathrm{Hom}(P, P')$, the square

$$\begin{array}{ccc} \mathrm{Hom}(\mathsf{Y}(a), P) & \xrightarrow{\alpha_{P,a}} & P(a) \\ {\scriptstyle \mathrm{Hom}(\mathsf{Y}(p),\sigma)}\downarrow & & \downarrow{\scriptstyle \sigma(p)} \\ \mathrm{Hom}(\mathsf{Y}(a'), P') & \xrightarrow[\alpha_{P',a'}]{} & P'(a') \end{array}$$

commutes up to homotopy.

Proof. See Egbert Rijke's proof of this at the HoTT website, which resembles many standard proofs from category theory. □

Corollary 4.0.8. *By setting $\mathsf{Y}(b) = P$, we have that $\mathrm{Hom}(\mathsf{Y}(a), \mathsf{Y}(b)) \simeq a \rightsquigarrow b$.*

5 Models of type theory

This section is a very brief and informal summary of certain research in the semantics of HoTT.

A category modeling intensional type theory will have the following features.

$$\text{initial object} \longleftrightarrow \bot$$

$$\text{terminal object} \longleftrightarrow *$$

$$\text{product} \longleftrightarrow P \times Q$$

$$\text{coproduct} \longleftrightarrow P + Q \quad .$$

$$\text{cartesian closed} \longleftrightarrow P \to Q$$

$$\text{locally cartesian closed} \longleftrightarrow \prod_{x:A} P(x)$$

$$\text{weak factorization system} \longleftrightarrow a = b$$

Any proof in our type theory produces a theorem in any model of it. It is known that the structure of Martin-Löf identity types is that of ∞-groupoids (i.e., ∞-categories in which any k-morphism is an equivalence).

HoTT without univalence may be modeled in a certain extension of ZFC. In 2006, Voevodsky showed that the category of Kan complexes (i.e., simplicial sets with Kan fillers) is a model of HoTT. In it, a type family corresponds to a Kan fibration. His construction, however, is non-constructive to some degree. Thus, it fails as a genuinely computational interpretation of HoTT.

Coquand, however, has designed a so-called cubical model of HoTT, in which the univalence axiom is a theorem. This models the ∞-groupoid structure of identity types with the category of constructive cubical sets, which are set-valued presheaves on the category of power sets of finite sets (viewed as n-cubes). Thus, Coquand's model is a constructive account of the univalence axiom.

Anders Mörtberg has designed an experimental implementation of cubical type theory, called Cubicaltt, which is on his GitHub page.

Further reading

A key feature of HoTT that we have neglected is the ability to define so-called higher inductive types (HIT's). For example, the circle S^1 can be modeled as a HIT, from which we can prove that $\pi_1(S^1) \cong \mathbb{Z}$. Dan Licata has a formalized proof of this in Agda.

Sources

1. *nLab*.
2. https://homotopytypetheory.org/.
3. The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. First edition, 2013.
4. Altenkirch, Thorsten. "Introduction to Homotopy Type Theory." *EWSCS 2017*. 2017.