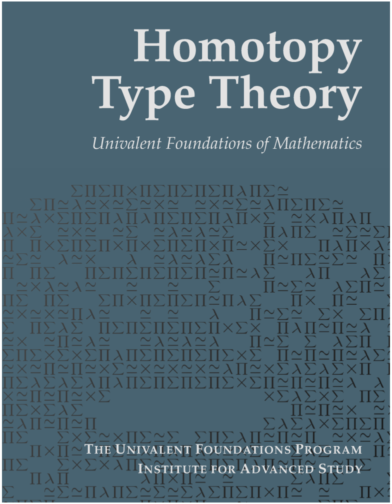
Recent progress in Homotopy type theory

Univalent Foundations team

July 22nd, 2013

Supported by EU FP7 STREP FET-open ForMATH

Most of the presentation is based on the book:



Homotopy type theory

Collaborative effort lead by Awodey, Coquand, Voevodsky
at Institute for Advanced Study
Book, library of formal proofs (Coq, agda).

Towards a new practical foundation for mathematics.
Closer to mathematical practice, inherent treatment of equivalences.

Towards a new design of proof assistants:
Proof assistant with a clear (denotational) semantics, guiding the addition of new features.

Concise computer proofs (deBruijn factor < 1 !).

Challenges

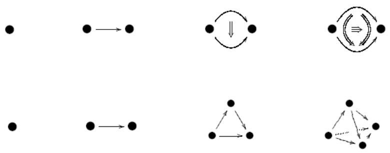
Sets in Coq setoids, no unique choice (quasi-topos), ...

Coq in Sets somewhat tricky, not fully abstract (UIP,...)

Towards a more symmetric treatment.

Two generalizations of Sets

To keep track of isomorphisms we want to generalize sets to groupoids (categories with all morphisms invertible),
(proof relevant equivalence relations)
2-groupoids (add coherence conditions for associativity),
. . . , ∞-groupoids
∞-groupoids are modeled by Kan simplicial sets.
(Grothendieck homotopy hypothesis)



Topos theory

Wikimedia Commons

Topos theory

A topos is like:

- a semantics for intuitionistic formal systems
  model of intuitionistic higher order logic.
- a category of sheaves on a site
- a category with finite limits and power-objects
- a generalized space

Higher topos theory

A higher topos is like:

- a semantics for Martin-Löf type theory with univalence and higher inductive types ??
- a model category which is Quillen equivalent to simplicial $PSh(C)_S$ for some model site (C, S).
- a generalized space (presented by homotopy types)
- a place for abstract homotopy theory
- a place for abstract algebraic topology

Envisioned applications

Type theory with univalence and higher inductive types as the internal language for higher topos theory?

- higher categorical foundation of mathematics
- framework for formalization of mathematics
  internalizes reasoning with isomorphisms
- expressive programming language
- language for synthetic pre-quantum physics (like Bohrification)
  Schreiber/Shulman

Here: develop mathematics in this framework.
Partial realization of Grothendieck's dream:
axiomatic theory of ∞-groupoids.

Homotopy Type Theory

The homotopical interpretation of type theory is that we think of:

- types as spaces
- dependent types as fibrations (continuous families of types)
- identity types as path spaces

We define homotopy between functions $A \to B$ by:
$f \sim g :\equiv \prod_{(x:A)} f(x) =_B g(x)$.
The function extensionality principle asserts that the canonical function $(f =_{A \to B} g) \to (f \sim g)$ is an equivalence.

(homotopy type) theory = homotopy (type theory)

The hierarchy of complexity

Definition

We say that a type A is contractible if there is an element of type

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

Contractible types are said to be of level -2.

Definition

We say that a type A is a mere proposition if there is an element of type

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

Mere propositions are said to be of level -1.

The hierarchy of complexity

Definition

We say that a type A is a set if there is an element of type

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

Sets are said to be of level 0.

Definition

Let A be a type. We define

$$\text{is-}(-2)\text{-type}(A) :\equiv \mathsf{isContr}(A)$$
$$\text{is-}(n+1)\text{-type}(A) :\equiv \prod_{x,y:A} \text{is-}n\text{-type}(x =_A y)$$

Equivalence

A good (homotopical) definition of equivalence is:

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

This is a mere proposition.

The classes of n-types are closed under

- dependent products
- dependent sums
- idenity types
- W-types, when $n \geq -1$
- equivalences

Thus, besides ‘propositions as types’ we also get propositions as n-types for every $n \geq -2$. Often, we will stick to ‘propositions as types’, but some mathematical concepts (e.g. the axiom of choice) are better interpreted using ‘propositions as (-1)-types’.

Concise formal proofs

The identity type of the universe

The univalence axiom describes the identity type of the universe Type. There is a canonical function

$$(A =_{\mathsf{Type}} B) \to (A \simeq B)$$

The univalence axiom: this function is an equivalence.

- The univalence axiom formalizes the informal practice of substituting a structure for an isomorphic one.
- It implies function extensionality
- It is used to reason about higher inductive types

Voevodsky: The univalence axiom holds in Kan simplicial sets.

Direct consequences

Univalence implies:

- functional extensionality
- logically equivalent propositions are equal
  Lemma uahp ‘{ua:Univalence}: forall P P’: hProp, (P $\leftrightarrow$ P’)$\rightarrow$ P = P’.
- isomorphic Sets are equal
  all definable type theoretical constructions respect isomorphisms

Theorem (Structure invariance principle)

Isomorphic structures (monoids, groups,...) may be identified.

Informal in Bourbaki. Formalized in agda (Coquand, Danielsson).

HITs

Higher inductive types were conceived by Bauer, Lumsdaine, Shulman and Warren.

The first examples of higher inductive types include:

- The interval
- The circle
- Propositional reflection

It was shown that:

- Having the interval implies function extensionality.
- The fundamental group of the circle is $\mathbb{Z}$.

Higher inductive types internalize colimits.

Higher inductive types

Higher inductive types generalize inductive types by freely adding higher structure (equalities).
Preliminary proposal for syntax (Shulman/Lumsdaine).
Impredicative encoding of some HITs,
like initial implementation of inductive types in Coq.
Can be introduced using axioms, does not compute.
Experimental work: use modules (in agda),
similar technology has been implemented by Bertot in Coq.

With higher inductive types, we allow paths among the basic constructors. For example:

- The interval I has basic constructors

$$0_I, 1_I : I \qquad \text{and} \qquad \mathsf{seg} : 0_I =_I 1_I.$$

- The circle $\mathbb{S}^1$ has basic constructors

$$\mathsf{base} : \mathbb{S}^1 \qquad \text{and} \qquad \mathsf{loop} : \mathsf{base} =_{\mathbb{S}^1} \mathsf{base}.$$

With paths among the basic constructors, the induction principle becomes more complicated.

Squash

NuPrl's squash equates all terms in a type

Higher inductive definition:

```
Inductive minus1Trunc (A : Type) : Type :=
  | min1 : A → minus1Trunc A
  | min1_path : forall (x y: minus1Trunc A), x = y
```

Reflection into the mere propositions

Logic

Set theoretic foundation is formulated in first order logic.
In type theory logic can be defined, propositions as (-1)-types:

$$
\begin{aligned}
\top &:\equiv \mathbf{1} \\
\bot &:\equiv \mathbf{0} \\
P \wedge Q &:\equiv P \times Q \\
P \Rightarrow Q &:\equiv P \to Q \\
P \Leftrightarrow Q &:\equiv P = Q \\
\neg P &:\equiv P \to \mathbf{0} \\
P \vee Q &:\equiv \| P + Q \| \\
\forall (x : A).\, P(x) &:\equiv \prod_{x:A} P(x) \\
\exists (x : A).\, P(x) &:\equiv \left\| \sum_{x:A} P(x) \right\|
\end{aligned}
$$

models constructive logic, not axiom of choice.

Unique choice

```
Definition hexists {X} (P:X→ Type):=(minus1Trunc (sigT P) ).

Definition atmost1P {X} (P:X→ Type):=
    (forall x1 x2 :X, P x1 → P x2 → (x1 = x2 )).

Definition hunique {X} (P:X→ Type):=(hexists P) * (atmost1P P).

Lemma iota {X} (P:X→ Type):
  (forall x, IsHProp (P x)) → (hunique P) → sigT P.
```

In Coq we cannot escape Prop.

Basic properties

Lemma

Suppose $P : A \to \text{Type}$ *is a family of types, let* $p : x =_A y$ *and let* $u : P(x)$*. Then there is a term* $p_*(u) : P(y)$*, called the transportation of* u *along* p*.*

Lemma

Suppose $f : \prod_{(x:A)} P(x)$ *is a dependent function, and let* $p : x =_A y$*. Then there is a path* $f(p) : p_*(f(x)) =_{P(y)} f(y)$*.*

In the case of the interval, we see that in order for a function $f : \prod_{(x:I)} P(x)$ to exist, we must have

$$\begin{aligned} f(0_I) &: P(0_I) \\ f(1_I) &: P(1_I) \\ f(\mathsf{seg}) &: \mathsf{seg}_*(f(0_I)) =_{P(1_I)} f(1_I) \end{aligned}$$

Interval

```
Module Export Interval.
Local Inductive interval : Type :=
  | zero : interval
  | one : interval.
Axiom seg : zero = one.

Definition interval_rect (P : interval → Type)
  (a : P zero) (b : P one) (p : seg # a = b)
  : forall x:interval, P x
  := fun x ⇒ match x return P x with
                | zero ⇒ a
                | one ⇒ b
              end.

Axiom interval_rect_beta_seg : forall (P : interval → Type)
  (a : P zero) (b : P one) (p : seg # a = b),
  apD (interval_rect P a b p) seg = p.
End Interval.
```

discriminate is disabled.

Induction with the interval

The induction principle for the interval is that for every $P : I \to \text{Type}$, if there are

- $u : P(0_I)$ and $v : P(1_I)$
- $p : \text{seg}_*(u) =_{P(1_I)} v$

then there is a function $f : \prod_{(x:I)} P(x)$ with

- $f(0_I) :\equiv u$ and $f(1_I) :\equiv v$
- $f(\text{seg}) = p$.

Induction with the circle

The induction principle for the circle is that for every $P : \mathbb{S}^1 \to \mathsf{Type}$, if there are

- $u : P(\mathsf{base})$
- $p : \mathsf{loop}_*(u) =_{P(\mathsf{base})} u$

then there is a function $f : \prod_{(x:\mathbb{S}^1)} P(x)$ with

- $f(\mathsf{base}) :\equiv u$
- $f(\mathsf{loop}) = p$.

Using univalence to reason about HITs

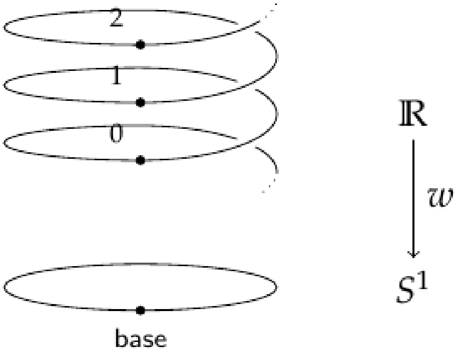
How do we use univalence to reason about HITs?

- Suppose we have a HIT W.
- and we want to describe a property $P : W \to \text{Type}$.
- for the point constructors of W we have to give types.
- for the path constructors of W we have to give paths between those types
- by univalence, it suffices to give equivalences between those types.

Suppose, in our inductive type W we have $p : x =_W y$ and $P(x) :\equiv A$ and $P(y) :\equiv B$ and to p we have assigned the equivalence $e : A \simeq B$.

Then transporting along p computes as applying the equivalence e.

The universal cover, computing base $=_{\mathbb{S}^1}$ base



The universal cover, computing base $=_{\mathbb{S}^1}$ base

Licata/Shulman: With this idea, we can construct the universal cover of the circle: $C : \mathbb{S}^1 \to \mathsf{Type}$. Our goal is to use C to show

$$(\mathsf{base} =_{\mathbb{S}^1} \mathsf{base}) \simeq \mathbb{Z}.$$

We define $C : \mathbb{S}^1 \to \mathsf{Type}$ by:

- $C(\mathsf{base}) :\equiv \mathbb{Z}$
- To transport along loop we apply the equivalence $\mathsf{succ} : \mathbb{Z} \to \mathbb{Z}$.

Theorem

The cover C has the property that

$$\mathsf{isContr}\left(\sum_{(x:\mathbb{S}^1)} C(x)\right)$$

'$\mathbb{R}$ is contractible'

Before we prove the theorem let us indicate why it is useful.

- Suppose A, $a : A$ is a type and $P : A \to \text{Type}$.
- there is a term of $P(a)$.
- and $\sum_{(x:A)} P(x)$ is contractible.

Note that

- The singleton $\sum_{(x:A)} x =_A a$ is contractible
- by the assumption $P(a)$, there exists a function

$$f(x) : (x =_A a) \to P(x)$$

for every $x : A$.

Theorem

If $f : \prod_{(x:A)} P(x) \to Q(x)$ *induces an equivalence*

$$(\sum_{(x:A)} P(x)) \to (\sum_{(x:A)} Q(x)),$$

then each $f(x) : P(x) \to Q(x)$ *is an equivalence.*

Hence under the above assumptions we obtain that

$$P(x) \simeq (x =_A a)$$

In particular, the theorem about the universal cover has the corollary that

$$C(x) \simeq (x =_{\mathbb{S}^1} \text{base})$$

Theorem

The cover C has the property that

$$\mathsf{isContr}\left(\sum_{(x:\mathbb{S}^1)} C(x)\right)$$

$(\mathsf{base}; 0)$ is the center of contraction and

$$\alpha : \prod_{(k:\mathbb{Z})} \sum_{(p:\mathsf{base}=_{\mathbb{S}^1}\mathsf{base})} p_*(k) =_{\mathbb{Z}} 0.$$

With some calculations:

Theorem

$(\mathsf{base} =_{\mathbb{S}^1} \mathsf{base}) \simeq \mathbb{Z}$.

Fundamental group of the circle is $\mathbb{Z}$.
The proof is by induction on $\mathbb{S}^1$.

Formal proofs

This theorem has a Coq/agda proof.
Likewise, the following has been done:

- total space of Hopf fibration
- computing homotopy groups upto $\pi_4(S^3)$
- Freudenthal suspension theorem
- van Kampen theorem
- James construction
- ...

Most proofs are formalized, with short proofs.

Quotients

Towards sets in homotopy type theory.

Voevodsky: univalence provides quotients.

Quotients can also be defined as a higher inductive type

```
Inductive Quot (A : Type) (R:rel A) : hSet :=
  | quot : A → Quot A
  | quot_path : forall x y, (R x y), quot x = quot y
(* | _ :isset (Quot A).*)
```

Truncated colimit.

We verified the universal properties of quotients.

Modelling set theory

Theorem (Rijke,S)

*0-*Type *is a* ΠW*-pretopos (constructive set theory).*

Assuming AC, we have a well-pointed boolean elementary topos with choice (Lawvere set theory).
Define the cumulative hierarchy $\emptyset, P(\emptyset), \dots, P(V_\omega), \dots,$
by higher induction. Then V is a model of constructive set theory.

Theorem

Assuming AC, V models ZFC.

We have retrieved the old foundation.

Subobject classifier

$$
\begin{array}{ccc}
I & \xrightarrow{\;!\;} & 1 \\
{\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle \mathsf{True}} \\
A & \xrightarrow{\;P\;} & \mathsf{Prop}
\end{array}
$$

With propositional univalence, hProp classifies monos into A.
Equivalence between predicates and subsets.
This correspondence is the crucial property of a topos.

Object classifier

$Fam(A) := \{(I, \alpha) \mid I : Type, \alpha : I \to A\}$ (slice cat)
$Fam(A) \cong A \to Type$
(Grothendieck construction, using univalence)

$$\begin{array}{ccc} I & \xrightarrow{i} & \mathrm{Type}_\bullet \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\pi_1} \\ A & \xrightarrow{P} & \mathrm{Type} \end{array}$$

$\mathrm{Type}_\bullet = \{(B, x) \mid B : \mathrm{Type}, x : B\}$
Classifies *all* maps into A + group action of isomorphisms
Crucial construction in ∞-toposes.
Proper treatment of Grothendieck universes from set theory.
Formalized in Coq. Induced improved treatment of universe polymorphism.

1-Category theory

Type of objects. Hom-set (0-Type) between any two elements.
Isomorphic objects objects are equal.
'Rezk complete categories.'

Theorem

$F : A \to B$ is an equivalence of categories iff it is an isomorphism.

Generalization of the Structure Identity Principle

Every pre-category has a Rezk completion.
Formalized in Coq (Ahrens, Kapulkin, Shulman).

Towards higher topos theory

Rijke/S/Shulman are developing internal higher topos theory.

- Factorization systems for n-levels, generalizing epi-mono factorization.
- Modal type theory for reflective subtoposes, sheafification.
- Homotopy colimits by higher inductive types behave well (descent theorem), using an internal model construction: graph presheaf model of type theory.

Computational interpretation

Coquand: Kan semisimplicial set model in type theory without Id-types gives an a priori computational interpretation of univalence and HITs.
A more operational interpretation (for groupoids) by Harper-Licata.
In fact, these reductions (push through the isomorphisms) suggests new proofs in algebraic topology.

Conclusion

Book, library of formal proofs.

Towards a new practical foundation for mathematics based on higher topos theory.
Closer to mathematical practice, less ad hoc encodings.

Towards a new design of proof assistants:
Proof assistant with a clear semantics,
guiding the addition of new features.

homotopytypetheory.org