An introduction to Homotopy Type Theory

Bruno Barras

October 18, 2013

Overview

Genesis of HoTT

Observation (Hofmann & Streicher): in intensional Martin-Löf's Type Theory, $(X, =_X)$ has a groupoid structure.

In 2005, Voevodsky and Awodey independently realized that MLTT was the language of choice for formalizing homotopy theory.

Synthetic Homotopy Theory

- Spaces are represented by types:

  X is a space $\qquad \vdash X : \mathrm{Type}$

- Points of a space are its inhabitants:

  a is a point of X $\qquad \vdash a : X$

- Paths are witnesses of equality:

  p is a path from a to b in X $\qquad \vdash p : a =_X b$

- Homotopies are witnesses of equality between paths:

  q is an homotopy between paths p and p' in X

  $$\vdash q : p =_{a =_X b} p'$$

- etc.

Mismatches with usual Type Theory

- Equality is not a proposition (possibly proof irrelevant) anymore

$$\frac{\vdash X : \mathrm{Type}_i \quad \vdash a : X \quad \vdash b : X}{\vdash a =_X b : \mathrm{Type}_i}$$

- Singleton elimination (strong elimination for $=$) would make the above change useless.
- Uniqueness of Identity Proofs (UIP or K) is inconsistent with the HoTT interpretation.

$\Rightarrow$ The typing rules of equality (and in general: inductive types with indices) have to be restricted, which invalidates singleton elimination.

Univalence axiom

Univalence is a principle that allows to prove that 2 given spaces are homotopically equivalent.

It can be viewed as a strong form of extensionality (see later). Remember:

- Functional extensionality:

$$f = g \quad \text{iff} \quad \forall x.\, f(x) = g(x)$$

- Propositional extensionality:

$$A = B \quad \text{iff} \quad A \rightarrow B \wedge B \rightarrow A$$

- Reasoning up to isomorphism (in Type Theory, no principle lets us discriminate between isomorphic types):

$$A = B \quad \text{iff} \quad \exists f\, g.\, f \circ g = 1 \ \wedge \ g \circ f = 1$$

Univalence: weak equivalences

Captures the notion of homotopy equivalent spaces.

$f : A \to B$ is a weak equivalence (between A and B) is a structure of:

- An inverse of f

$$g : B \to A$$

- g is the inverse of f:

$$r \colon \Pi a{:}A.\, g(f(a)) =_A a$$
$$s \colon \Pi b{:}B.\, f(g(b)) =_B b$$

- a coherence condition:

$$\Pi a{:}A.\, f(r(a)) =_{f(g(f(a)))=f(a)} s(f(a))$$

Univalence axiom

[Notation: $A \simeq B$ is a couple of a $f : A \to B$ and a proof that f is a weak equivalence.]

- Simplified statement: $(A = B) \simeq (A \simeq B)$
  (equality between types is weakly equivalent to weak equivalence)
- More precisely: the obvious function $A = B \to A \simeq B$ is a weak equivalence.
  In particular, we have: $A \simeq B \to A = B$.

Univalence contradicts UIP: there are 2 weak equivalences between `bool` and `bool` (identity and negation).

hoqtop : an implementation of HoTT

Github repository HoTT/coq and its companion standard library HoTT/HoTT.

Features:

- Option `-indices-matter` disables singleton elimination and puts equality at the Type level.
- Universe polymorphism.
- Univalence is assumed.
- Higher Inductive Types (HITs).

HoTT as a foundational formalism

Questions:

- Can we encode all of the theorems of the “standard” foundation in HoTT (maybe by assuming further consistent axioms) ?
- How can we reconcile UIP and univalence ?
- Are the extra features of HoTT of pratical interest for general use?

Homotopy Level

Classification of types according to their "dimension":

- Contractible types (level -2):

$$\mathrm{Contr}(X) := \Sigma c{:}X.\, \Pi a{:}X.\, a = c$$

- Type X has level $n+1$ if $a =_X b$ has level n for all a, b.

(Note: not all types need to have an h-level!)

Levels of particular interest:

- (-1): propositions
  (proof-irrelevance, at most one connected and contractible component)
- (0): sets, setoids
  (UIP holds for sets)
- (1): groupoids

Degenerated forms of univalence

Remember: $A \simeq B$ is

- $f : A \to B$
- $g : B \to A$
- $r : \forall a.g(f(a)) =_A a$
- $s : \forall b.f(g(b)) =_B b$
- $\forall a.f(r(a)) = s(f(a))$

When A and B are propositions , $A \simeq B$ amounts to $(A \to B) \times (B \to A)$.

- We have propositional extensionality.

When A and B are sets, $A \simeq B$ amounts to an isomorphism

- We have reasoning up to isomorphism.

Univalence + interval (see HITs, later) implies functional extensionality.

Covering all “set”-level maths

- The Set class of types is closed under usual type-theoretic constructions (0, 1, 2, Σ, Π, W-types)
- So, we recover set-level maths by constraining all manipulated types to be sets.

What have we gained ?

Relevant mathematics:

- A formal clarification of the distinction between Σ and $\exists$.
- Already familiar for educated Coq users.

Reasoning up to isomorphism:

- $\texttt{neg}$ leads to a proof of $\texttt{bool} = \texttt{bool}$

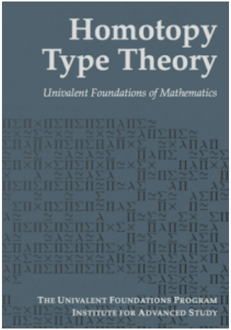
  $$J(\lambda X.\,\texttt{bool} \to X \to X,\ \texttt{neg},\ \texttt{and}) = \lambda b\, b'.\texttt{neg}(\texttt{and}(b,\ \texttt{neg}(b')))$$

- Transport of structures:
  e.g. monoid signature: $\Sigma X : \mathrm{Type}.\ \Sigma 1 : X.\ X \to X \to X$.

(More to come with HITs).

HoTT Book

Explain all this to regular mathematicians.



Freely downloadable from
`http://homotopytypetheory.org/book/`

Overview

Models

Two non-constructive models of HoTT:

- Geometric realization (Warren)
  $a =_X b$ is $f : [0;1] \to X$ (continuous, with $f(0) = a$, $f(1) = b$)
- Simplicial sets (Voevodsky)

Simplicials Set in Set Theory

Decomposition of a space in points, edges, surfaces, etc.

- a sequence of sets $(X_n)_{n\in\mathbb{N}}$
- face maps: $d_i^n : X_n \to X_{n-1}$ (for $0 \leq i \leq n$)
  d_i access the face of lower dimension that does not contain the i-th point.
- face map conditions: $d_j \circ d_i = d_i \circ d_{j+1}$ (when $i \leq j$)
- degeneracy maps: $s_i^n : X_n \to X_{n+1}$ (for $0 \leq i < n$) s_i is the degenerated simplex where the i-th point has been repeated.
- degeneracy map conditions...

Kan completions

- Kan completions: any “horn” can be completed and filled.
- Model based on Kan complexes.

Effectivity issue with dependent product: needs decidability of degeneracies.

Simplicial Set in Type Theory

- Following the set-theoretical definition would be awkward (rewriting)

Semi-simplicial types:

- $X_0 : \mathrm{Type}$
- $X_1 : X_0 \to X_0 \to \mathrm{Type}$
- $X_2 : \Pi a_0 : X_0 . \Pi a_1 : X_0 . \Pi a_2 : X_0 .\ X_1(a_0, a_1) \to X_1(a_0, a_2) \to X_1(a_1, a_2) \to \mathrm{Type}$
- etc.

Face maps are not needed (faces are defined up to definitional equality).

Hard to define the general case! (Herbelin)

Setoid model

A 1-truncated semi-simplicial type:

- A couple (X_0, X_1) as before,
- equipped with level 0 completion and filling, and level 1 completion

is equivalent to a setoid:

- a type and a relation
- a proof that the relation is an equivalence

Generalizes (better) to higher dimensions: 2-truncated SST correspond to groupoids.

In the above setoid model:

- (degenerated) univalence holds,
- the universe of setoids is a groupoid.

Overview

Need for a new primitive

Kraus has shown that Type_n is not of hlevel n.

In Coquand's model, Type_n has exactly hlevel $n+1$.

So, we cannot build a non-set type in Type_0.

HITs

A generalization of ususal inductive types:

- Possibility to give path constructors (not only point constructors).
- Elimination (pattern-matching) is restricted to ensure the preservation of equality.

Example: Circle

```
Inductive S1 :=
  base : S1
with paths :=
  loop : base=base.
```

Besides the above formation/introduction rules, the eliminator (match) has the following type:

```
fun P f g c =>
  match c return (P c) with
  | base => f
  | loop => g
  end
: forall (P:S1->Type) (f:P base),
    transp P loop f = f -> forall c:S1, P c
```

Using univalence, one can prove $(\texttt{base} = \texttt{base}) = \mathbb{Z}$.

Example: Interval

```
Inductive Interval :=
  left | right
with paths :=
  segment : left=right.
```

Using this definition and univalence, one can derive functional extensionality.

Example: Suspension

```
Inductive Susp (X : Type) : Type :=
    | north : Susp X
    | south : Susp X
with paths :=
    | merid (x:X) : north = south.

Definition S2 := Susp S1.
```

Impact on set-level maths

They should form a good way to represent quotients (once the computational interpretation of univalence is solved).

```
Inductive Z_2Z :=
  O | S (_:Z_2Z)
with paths :=
  mod2 : O = S (S O).
```

A similar definition

```
Inductive Z_2Z' :=
  O | S (_:Z_2Z')
with paths :=
  mod2 n : n = S (S n).
```

would produce a non-set, so truncation would be required.

Conclusions

Despite apparent contradiction with popular axioms (UIP), HoTT can be seen as a new foundation for mathematics.

Univalence and HITs may have a positive impact on the way everyday maths can be expressed.