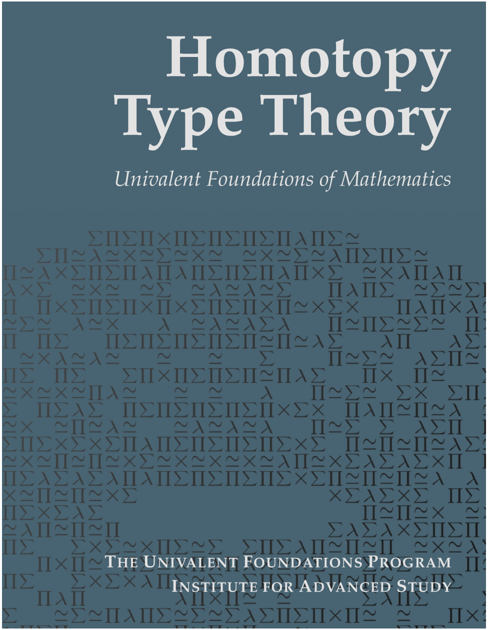
Formalizing mathematics in the univalent foundations

Bas Spitters

Nov 25h, 2014

About me

- PhD thesis on constructive functional analysis
- Connecting Bishop's pointwise mathematics with formal topology/topos th (w Coquand)
- Formalization of effective real analysis in Coq
  building on non-efficient corn library
  O'Connor's PhD, led WP in EU STREP-FET ForMath
- Topos theory and quantum theory
- Univalent foundations as a combination of these strands
  co-author of the book and the Coq library


Homotopy
Type Theory
Univalent Foundations of Mathematics
The Univalent Foundations Program
Institute for Advanced Study

Homotopy type theory

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.

Challenges

Sets in Coq setoids (no subsets, quotients), 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 (proof relevant equivalence relations)
2-groupoids (add coherence conditions for associativity),
..., ∞-groupoids

Categorical logic

Curry-Howard:
simply typed λ-calculus
cartesian closed categories
minimal logic

extensional dependent type theory
locally cartesian closed categories
predicate logic.

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

Combine these two generalizations.

A higher topos is like:

- a model category which is Quillen equivalent to simplicial $Sh(C)_S$ 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
- a semantics for Martin-Löf type theory with univalence and higher inductive types?

Rezk, Lurie

Higher topos theory

Prime example: Kan simplicial sets/∞-groupoids.
VV: HoTT+univalence is modeled in Kan sSets.
Shulman/Cisinski: HoTT+univalence for h-Tarski universes can be interpreted in any Grothendieck ∞-topos.

h=Hofmann, homotopy
Type U of codes. Coercion $\mathrm{El} : U \to \mathrm{Type}$, plus operations like

$$\mathrm{Pi} : \Pi a : U, (\mathrm{El} a \to U) \to U$$

El only respects these operations up to propositional equality:

$$\mathrm{El}(\mathrm{Pi} ab) = \Pi x : \mathrm{El} a, \mathrm{El}(bx)$$

Strict models over Reedy categories.

Elementary higher topos

Grothendieck topos: Sheaves on a site (formal topology)
Elementary topos (Lawvere-Tierney): abstract (logical) definition

Likewise:
Higher topos (Rezk, Lurie, ...)
Quest for an elementary higher topos (Awodey, Shulman, Joyal, ...)

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 large scale formalization of mathematics
- expressive programming language
- *higher* topos of trees (Birkedal, Møgelberg)
- synthetic pre-quantum physics
  (Schreiber/Shulman, cf. Bohr toposes)

Effective ∞-topos?, gluing (Shulman), sheaf models, ...
Partial realization of Grothendieck's dream:
(generalized) algebraic theory of ∞-groupoids.
Here: Develop mathematics in this framework
Fact: Many theorems from higher topos theory have direct analogues in type theory.
Coq formalization[2]

[2] https://github.com/HoTT/HoTT/

Homotopy Type Theory

The homotopical interpretation of type theory:

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

(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

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

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.

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.

The classes of n-types are closed under

- dependent products
- dependent sums
- identity 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 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.
Coquand etal: Computational interpretation in Kan cubical sets.
Implemented in haskell.

Direct consequences of Univalence

Univalence implies:

- functional extensionality
  ```
  Lemma ap10 {A B} (f g : A → B ): (f=g → f == g).
  Lemma FunExt {A B}: forall f g, IsEquiv (ap10 f g).
  ```
- logically equivalent propositions are equal:
  ```
  Lemma uahp `{ua:Univalence}: forall P P': hProp, (P ↔ P')→ 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).

Higher inductive types

Higher inductive types internalize colimits.
Higher inductive types generalize inductive types by freely adding higher structure (equalities).
Allows to develop much of algebraic topology synthetically.
Here we focus on generalized quotients.

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

Awodey, Bauer [ ]-types.

Theorem

epi-mono factorization. Set is a regular category.

Usual proof use impredicativity. Here: universe polymorphism.

Generalizes to all truncations.

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.
```

On the contrary, in Coq we cannot escape Prop.
Exact completion: add quotients to a category.
Similarly: Consider setoids $(T, \equiv)$.
Spiwack: Prop-valued Setoids in Coq give a quasi-topos.
In UF we have a topos.

Quotients

Towards sets in homotopy type theory.

Voevodsky: univalence provides (impredicative) 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.

These quotient types are predicative in Cub.

We verified the universal properties of quotients.

Modelling set theory

pretopos: extensive exact category
ΠW-pretopos: pretopos with Π and W-types.

Theorem

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

Assuming AC, a well-pointed boolean elementary topos with choice (=Lawvere set theory).

Predicativity

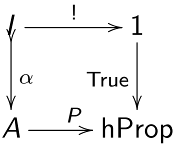
In predicative topos theory: no subobject classifier/power set.
Joyal/Moerdijk/Awodey/...: Algebraic Set Theory (AST).
AST provides a framework for defining various predicative toposes.
Categorical treatment of set and class theories.
Two challenges:

- From pure HoTT we do not (seem to) obtain the collection axiom from AST.
  Instead: Higher inductive types also provide free algebras.
- The universe is not a set, but a groupoid!

What is a higher categorical version of AST?
Perhaps HoTT already provides this...

Large subobject classifier

The subobject classifier lives in a higher universe.
Use universe polymorphism.



With propositional univalence, hProp classifies monos into A.
Equivalence between predicates and subsets.
This correspondence is the crucial property of a topos.
Sanity check: epis are surjective (by universe polymorphism).

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.
Improved treatment of universe polymorphism (h/t Sozeau).
Object classifier equivalent to univalence, assuming funext.

Conclusion

- Practical foundation for mathematics
- UF generalizes the old foundation
- Towards a proof assistant with a clear denotational semantics
  prototypes: Cubical, Andromeda, HTS
- Towards elementary higher topos theory