Mechanically verified mathematics in univalent foundations

Benedikt Ahrens

What is "mechanically verified" mathematics?

- Mathematics (definitions, statements, proofs) written in a formal language understood by a computer program: a **computer proof assistant**
- Correctness of proofs mechanically checked by the proof assistant:
  - Does the proof adhere to the rules determined by the foundations?
  - Does the proof prove the statement it claims to prove?

Why mechanical verification?

- Trust in a known system: the proof assistant's kernel
- To archive and disseminate knowledge in an interactive, searchable format
- Tool for teaching

Univalent foundations and proof assistants

Univalent foundations for proof assistants

- Voevodsky developed univalent foundations as a convenient foundation to mechanize mathematics in
- Voevodsky's starting point were proof assistants for Martin-Löf type theory, specifically the Coq proof assistant

Proof assistants for univalent foundations

- Ad-hoc changes to proof assistants for MLTT/CoC
  - Coq
  - Agda
- Development of new proof assistants with "native" univalence, based on cubical type theories

Outline

1. The UniMath library of mathematics in univalent style
    - The language underlying UniMath
    - The UniMath library
2. Things to do
    - Propositional resizing
    - Future work: Voevodsky's suggestions

Outline

Origin: Voevodsky's library `Foundations`

In Feb 2010, Voevodsky started writing the Coq library *Foundations*, making precise his ideas collected in *A very short note on homotopy λ-calculus*.

```
Fixpoint isofhlevel (n:nat) (X:UU): UU:=
match n with
O => iscontr X |
S m => forall x:X, forall x':X, (isofhlevel m (paths _ x x'))
end.


Theorem hlevelretract (n:nat)(X:UU)(Y:UU)(p:X -> Y)(s:Y ->X)(eps: forall y:Y, paths _  (p (s y)) y): (isofhlevel n X) -> (isofhlevel n Y).
Proof. intro. induction n. intros. apply (contrl1' _ _ p s eps X0).
intros. unfold isofhlevel. intros. unfold isofhlevel in X0. assert (is: isofhlevel n (paths _ (s x) (s x'))).  apply X0.
set (s':= maponpaths _ _ s x x'). set (p':= pathssec2 _ _ s p eps x x'). set (eps':= pathssec3 _ _ s p eps x x'). apply (IHn _ _ p' s' eps' is). Defined.

Corollary  hlevelweqf (n:nat)(X:UU)(Y:UU)(f:X -> Y)(is: isweq _ _ f): (isofhlevel n X) -> (isofhlevel n Y).
Proof. intros.  apply (hlevelretract n _ _ f (invmap _ _ f is) (weqfg _ _ f is)). assumption. Defined.

Corollary  hlevelweqb (n:nat)(X:UU)(Y:UU)(f:X -> Y)(is: isweq _ _ f): (isofhlevel n Y) -> (isofhlevel n X).
Proof. intros.  apply (hlevelretract n _ _ (invmap _ _ f is) f (weqgf _ _ f is)). assumption. Defined.


Definition isaprop (X:UU): UU := isofhlevel (S O) X.
```

Other libraries were built on top of *Foundations*.

Founding of the UniMath library

UniMath was founded in spring 2014, by combining three libraries:

- Foundations (Voevodsky)
- RezkCompletion (Ahrens, Kapulkin, Shulman) (started Feb 2013)
- Ktheory (Grayson) (started Oct 2013)

Outline

The language underlying UniMath, in theory

Type former	Notation	(special case)
Sigma type	$\sum_{x:A} B(x)$	$A \times B$
Product type	$\prod_{x:A} B(x)$	$A \to B$
Coproduct type	$A + B$	
Identity type	$a =_A b$	
Universes	$\mathsf{U}_0 : \mathsf{U}_1 : \mathsf{U}_2 : \ldots$	
Nat, Bool, 1, 0		

- Definitional η-rules for $\sum$ and $\prod$
- Axioms: function extensionality, univalence
- Resizing: any proposition lives in U_0

The language underlying UniMath, in theory

Type former	Notation	(special case)
Sigma type	$\sum_{x:A} B(x)$	$A \times B$
Product type	$\prod_{x:A} B(x)$	$A \to B$
Coproduct type	$A + B$	
Identity type	$a =_A b$	
Universes	$\mathsf{U}_0 : \mathsf{U}_1 : \mathsf{U}_2 : \ldots$	
Nat, Bool, 1, 0		

- Definitional η-rules for $\sum$ and $\prod$
- Axioms: function extensionality, univalence
- Resizing: any proposition lives in U_0
  Warning: not known to be consistent

The language underlying UniMath, in practice

A subset of the Coq language:

- no record types
- no inductive types
- no `match` construct

Coq features used to simulate the UniMath language:

- Avoid Coq's `Prop` for identity type by `-indices-matter` flag
- η for sums through primitive projections
- Resizing rule enabled by `-type-in-type` flag

The language underlying UniMath, in practice

A subset of the Coq language:

- no record types
- no inductive types
- no `match` construct

Coq features used to simulate the UniMath language:

- Avoid Coq's `Prop` for identity type by `-indices-matter` flag
- η for sums through primitive projections
- Resizing rule enabled by `-type-in-type` flag
  Warning: known to be inconsistent

Outline

1. The UniMath library of mathematics in univalent style
   - The language underlying UniMath
   - The UniMath library
2. Things to do
   - Propositional resizing
   - Future work: Voevodsky's suggestions

Some information on the UniMath library

- ca. 160,000 loc
- More repositories building on top of UniMath
  - TypeTheory (ca. 20,000 loc)
  - largecatmodules (ca. 10,000 loc)
  - SetHITs (ca. 7,000 loc)
- ca. 35 contributors, plus many maintenance contributions from Coq developers
- Distributed under free software license
- Available on `https://github.com/UniMath/UniMath`

The UniMath library

Organized in ‘packages’:

- Foundations
- Combinatorics
- Algebra
- Number Systems
- Synthetic Homotopy Theory
- Real Numbers
- Category Theory
- Homological Algebra
- K-theory
- Topology
- Homological Algebra
- Substitution Systems
- . . .

UniMath: what does it look like?

Demo—we look at:

- the encode-decode method for coproducts
- the proof that the type of types of hlevel n is of hlevel Sn

Outline

Outline

Summary: Propositional resizing

In idealized UniMath, there is a sequence $\mathsf{U}_0 : \mathsf{U}_1 : \mathsf{U}_2 : \mathsf{U}_3 : \ldots$ of universes.
In a talk at TYPES 2011, Voevodsky suggested a set of **resizing rules**, in particular:

- If type $A : \mathsf{U}_i$ is a proposition, then A lives in the lowest universe.
- For any universe U_i, the type $\mathsf{hProp}(\mathsf{U}_i) = \sum_{X:\mathsf{U}_i} \mathsf{isaprop}(X)$ lives in the lowest universe.

Weakened versions of those rules—“up to equivalence”—are validated by Voevodsky’s simplicial set model.

Propositional resizing axiom

Given $\mathtt{U} \leq \mathtt{U}'$ and

$$\mathtt{j} : \mathtt{hProp}\,\mathtt{U} \to \mathtt{hProp}\,\mathtt{U}'$$

postulate

```
rr1ax U U' : @isweq (hProp U) (hProp U') j
```

- Is compatible with Voevodsky's univalent model in simplicial sets and therefore is consistent modulo ZFC.

Propositional resizing rule

```
Gamma |- T : U
Gamma |- is : isaprop T
---------------------------------------
Gamma |- RR1(T, is) : U0

Gamma |- T : U
Gamma |- is : isaprop T
--------------------------------------
Gamma |- [El](RR1(T,is))==[El](T)
```

- Consistency of these rules with univalent type theory is unknown.

Use of resizing in (idealized) UniMath

Propositional resizing is needed to achieve that

- the propositional truncation of A,

$$||A|| :\equiv \prod_{P:\mathsf{hProp}(\mathsf{U})} (A \to P) \to P$$

  lives in the same universe as A

- the set quotient of (X, R) lives in the same universe as $X : \mathsf{U}_i$

  Note: elements of the quotient are equivalence classes

$$e : X \to \mathsf{hProp}(\mathsf{U}_k)$$

Research problems related to resizing

Show consistency of resizing rules in univalent type theory

In the TYPES 2011 talk, Voevodsky sketches a model of resizing rules that does not validate univalence.

Implement a proof assistant with propositional resizing

- In UniMath, resizing is currently achieved by the inconsistent rule $\mathsf{U} : \mathsf{U}$
- Dan Grayson is currently working on isolating the uses of $\mathsf{U} : \mathsf{U}$ into “resizing modules”

Outline

1. The UniMath library of mathematics in univalent style
   - The language underlying UniMath
   - The UniMath library
2. Things to do
   - Propositional resizing
   - Future work: Voevodsky's suggestions

Voevodsky's goals for UniMath

In a lecture in July 2017, Voevodsky outlined three goals for the UniMath library:

1. Mathematics of syntax and semantics of dependent type theories
2. Proof of Milnor's conjecture on Galois cohomology
3. Modern theory of geometry and topology of manifolds; in particular, construct a univalent category of smooth manifolds

Thanks for your attention.

References

- Voevodsky's emails to Dan Grayson
  https://groups.google.com/forum/#!topic/homotopytypetheory/K_4bAZEDRvE
- Voevodsky's library *Foundations*
  https://github.com/vladimirias/Foundations,
  archived at https://github.com/UniMath/Foundations
- Voevodsky's talk at TYPES 2011
  https://www.math.ias.edu/vladimir/sites/math.ias.edu.vladimir/files/2011_Bergen.pdf
- Voevodsky's talk on UniMath in July 2017
  https://www.newton.ac.uk/seminar/20170710113012301