Nov. 16, 2014, CMA2014, Kuwait.

Computer Proof Assistants and Univalent Foundations of Mathematics

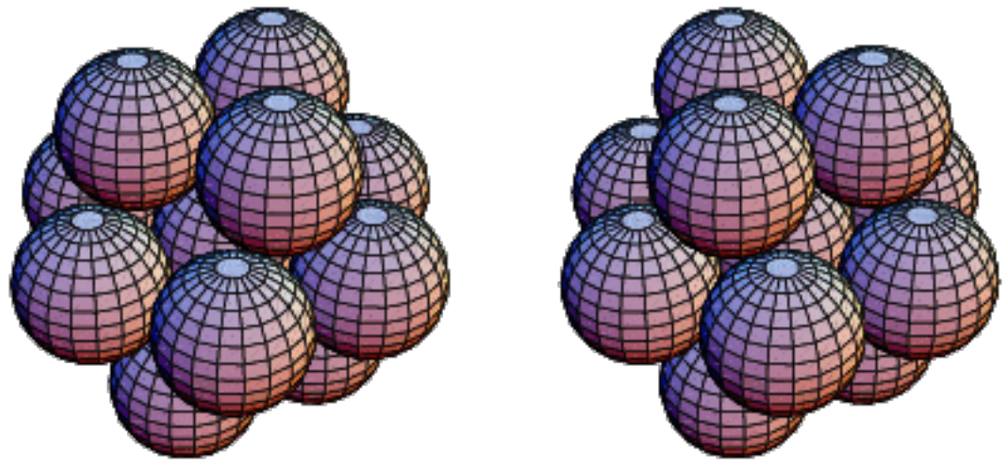
by Vladimir Voevodsky
from the Institute for Advanced Study in Princeton, NJ.

Kepler's Conjecture.

In 1611 Kepler proposed that the cubical and hexagonal packings of spheres in the 3d space are the densest possible ones.

The conjecture was proved by Tom Hales around 1997.

The proof was accepted for publication to Annals of Mathematics with a note from the reviewers that they were not able to check the proof.



From http://mathworld.wolfram.com/KeplerConjecture.html

Hales wanted certainty that his proof was correct.

Since the reviewers of his paper could not provide him with such certainty, he initiated, in 2003, "The Flyspeck Project". The goal of the project was to formalize the proof of Kepler's Conjecture, to have the formalized proof verified by a computer proof verification system and through this to achieve the certainty in its correctness.

Less than a month ago, on August 10, 2014, Tom Hales and his group announced the completion of the Flyspeck Project.

You can find a detailed and informative announcement of the completion on the Flyspeck Project webpage.

The mathematics of Hales's proof is formalized for the Flyspeck Project using a formalization system based on a slightly modified version of Church's formulation of the simple theory of types which Church published in 1940.

Most of the time and effort is taken by formalization of linear and nonlinear inequalities involving real numbers.

Here is an example of the kind of inequalities they had to deal with:

$$-1.44 < x_6x_2^2 + x_5x_3^2 - x_1x_4^2 + x_4^2 - \tfrac{1}{3}x_1 + \tfrac{4}{3}x_4$$
$$(x_1, x_2, x_3, x_4, x_5, x_6) \in [(-1, -0.1, -0.1, -1, -0.1, -0.1), (0, 0.9, 0.5, -0.1, -0.05, -0.03)]$$

If you look at the Church's 1940 paper and then at the inequality on the previous page you can guess the amount of work which has to be done to translate even one such inequality (and there were thousands!) into the formal language of Church's Type Theory.

The translation of the statements (e.g. inequalities) into the formal language and the translation of the proofs into the steps of formal inference was done with the help of two remarkable proof assistants HOL Light and Isabelle HOL.

HOL Light is authored and maintained by John Harrison a computer scientist and mathematician working at Intel.

Isabelle HOL was developed and is maintained by researchers at several european universities.

Both HOL Light and Isabelle are LCF-style proof assistants.

HOL is an abbreviation for "higher order logic" which is the name of the modified version of Church's type theory which these proof assistants use.

HOL is a classical formal system using heavily the Law of Excluded Middle and Axiom of Choice.

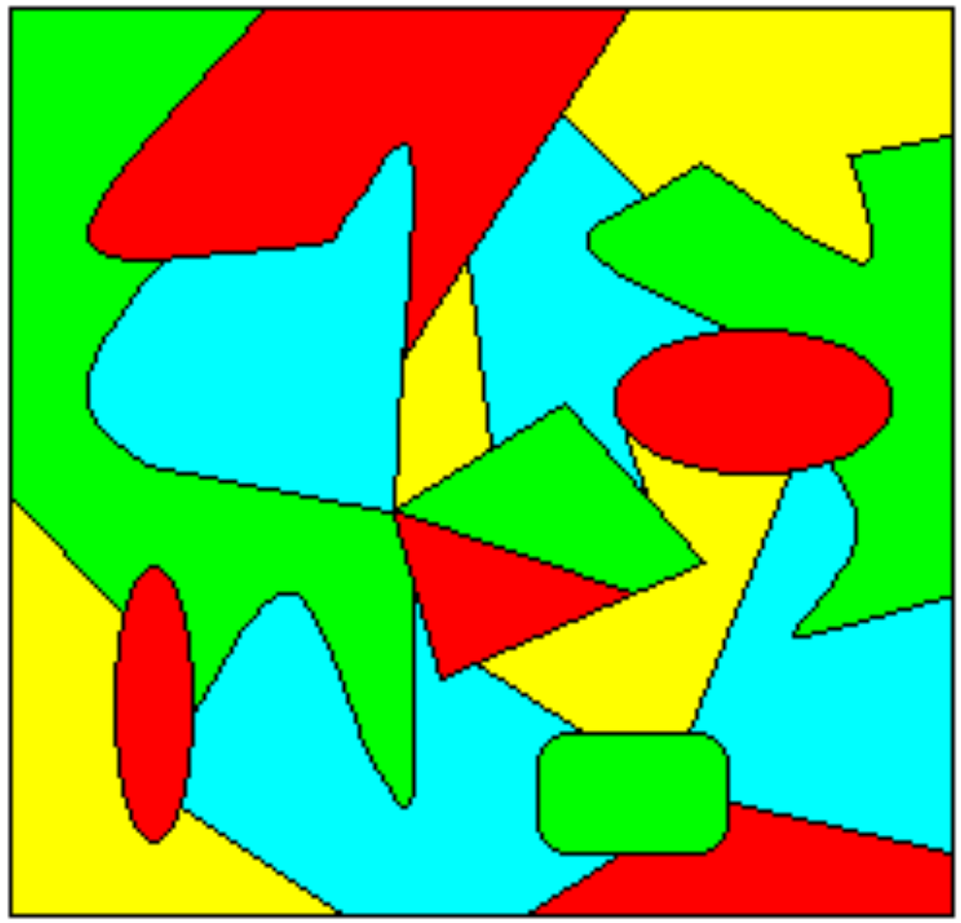
HOL provides an adequate foundations for arithmetic, real analysis and point-set topology.

It is not convenient for more recent mathematics such as homotopy theory or abstract algebra.

The Four Color Theorem.

The four color theorem was conjectured in 1852 by Francis Guthrie. It was proved in 1976 by Appel and Haken.

In 2005 Georges Gonthier completed a formalization of a proof of the four color theorem using proof assistant Coq.



From http://www.mathpages.com/home/kmath266/kmath266.htm

```
Variable R : real_model.
Theorem four_color : (m : (map R))
  (simple_map m) -> (map_colorable (4) m).
Proof.
Exact (compactness_extension four_color_finite).
Qed.
```

This is how the final theorem of Gonthier's formalization looks like.

"real_model" here is a model of real numbers, so properly speaking the theorem in the form proved by Gonthier reads as follows:

"For any model R of the theory of real numbers and any simple map (of the sphere) defined relative to this model there is given a coloring of this map with at most four colors."

Feit-Thompson Theorem.

The Feit-Thompson Theorem also known as Odd-Order Theorem asserts that any finite group of odd order is solvable or, equivalently, that any finite group of odd order has a proper non-trivial normal subgroup.

This result was first proved in 1963 by Walter Feit and John Griggs Thompson.

In 2012 a group led by Georges Gonthier completed the formalization and formal verification of a proof of the Odd-Order Theorem using proof assistant Coq.

The proof of the Feit-Thompson Theorem created by the Gonthier group is **fully constructive**. As such it is really a new proof of the theorem.

The proof assistant Coq was developed at INRIA, the French National Research Institute for Computer Science and Applied Mathematics.

It is based on a formal system that is called Calculus of Inductive Constructions (CIC) and that is a substantially extended version of the Martin-Lof Type Theory or MLTT.

The first variant of MLTT was created by Per Martin-Lof around 1972 as a new foundational system for constructive mathematics.

Calculus of Constructions was created around 1986 by Thierry Coquand and at about the same time Coquand started the development of Coq.

For many years CIC remained known only to people interested in constructive mathematics and computer scientists. The main reason for it was that it lacked natural semantics. While many models of CIC were constructed none of these models looked "canonical" and none of them explained one of the most prominent features of CIC and other versions of the Martin-Lof Type Theory.

This feature is that equality in these systems is expressed through a construction that associates to any type T and a pair of objects a b : T of T a type Id T a b together with some additional data and to prove equality between a and b in T one is required to construct an object of Id T a b.

But then, given two such proofs p1 and p2 one may consider the secondary identity type Id (Id T a b) p1 p2. What is the meaning of this type?

The first model where these secondary identity types where non-trivial was constructed by Martin Hofmann and Thomas Strechier in 1998. Further models with similar properties were constructed by Michael Warren, Richard Garner and Benno van den Berg.

But all these models lacked a convincing interpretation of another important feature of the Martin-Lof Type Theories - the universes. In Martin-Lof Type Theories one is allowed to quantify over objects such as sets, groups etc. To avoid known paradoxes one uses the mechanism of universes, non unlike Grothendieck universes.

But how the "canonical" semantics should interpret a universe? The answer to this question was given in my univalent model that was announced in 2009. The univalent model also provided a natural interpretation of higher identity types and of all other features of CIC.

The standard univalent models interpret types (the name for collections of objects in Martin-Lof type theory) not as sets but as objects of some model category that models the ordinary homotopy category. One can say that these models interpret types as homotopy types.

The univalent foundations say that mathematics may be considered as a study of structures on homotopy types.

The usual set-level mathematics studies structures on sets, i.e., on homotopy types of discrete topological spaces.

The category-level mathematics studies structures on homotopy 1-types. All such homotopy types can be described by groupoids and in the univalent foundations categories are considered to be groupoids with an additional structure.

Through the use of the univalent model, CIC and similar to it type theories provided a formal environment for the Univalent Foundations.

That the Univalent Foundations together with this environment can be used for practical formalization of mathematics have been demonstrated by a library for Coq that I wrote in 2009-2011 and that is now a part of a large library called UniMath. It is freely available on GitHub and can be easily located through web search.

The following slide is an example of a construction from UniMath.

```
647 Lemma rigtorngrdistr ( X : rig ) : isrdistr ( rigtorngop1 X ) ( rigtorngop2 X ) .
648 Proof . intro . unfold isrdistr .
649  apply ( setquotuniv3prop ( eqrelrigtorng X ) ( fun x x' x'' : rigtorngcarrier X => eqset (rigtorngop2 X (rigtorngop1 X x x') x'' )
650  (rigtorngop1 X (rigtorngop2 X x x'' ) (rigtorngop2 X x' x'' )) ) ) . intros x x' x'' . change ( paths ( setquotpr (eqrelrigtorng X )
651 ( rigtorngop2int X ( rigtorngop1int X x x' ) x'' ) ) ( setquotpr (eqrelrigtorng X ) ( rigtorngop1int X ( rigtorngop2int X x x'' )
652 ( rigtorngop2int X x' x'' ) ) ) ) . apply ( maponpaths ( setquotpr ( eqrelrigtorng X ) ) ) .  unfold rigtorngop1int .
653 unfold rigtorngop2int . simpl . set ( rd := rigrdistr X ) .  set ( cm1 := rigcomm1 X ) . set ( rr := abmonoidoprer ( rigop1axs X ) ) .
654   apply pathsdirprod .
655
656 rewrite ( rd _ _ ( pr1 x'' ) ) . rewrite ( rd _ _ ( pr2 x'' ) ) . apply ( rr _ ( pr1 x' * pr1 x'' ) (pr2 x * pr2 x'' ) _ ) .
657 rewrite ( rd _ _ ( pr1 x'' ) ) . rewrite ( rd _ _ ( pr2 x'' ) ) . apply ( rr _ (pr1 x' * pr2 x'' ) ( pr2 x * pr1 x'' ) _ ) . Defined .
658
659 Opaque rigtorngrdistr .
660
661 Definition rigtorngdistr ( X : rig ) : isdistr  ( rigtorngop1 X ) ( rigtorngop2 X )  :=
662 dirprodpair ( rigtorngldistr X ) ( rigtorngrdistr X ) .
663
664 Definition rigtorng ( X : rig ) : rng .
665 Proof . intro . split with ( @setwith2binoppair ( rigtorngcarrier X ) ( dirprodpair ( rigtorngop1 X ) ( rigtorngop2 X ) ) ) .
666 split . apply ( dirprodpair ( rigtorngop1axs X ) ( rigtorngismonoidop2 X ) ) . apply ( rigtorngdistr X ) .  Defined .
667
668
```

The last "Definition" here is the construction of the ring of differences of a rig where a rig is the same as a ring without subtraction such that for all x, one has 0*x=x*0=0.

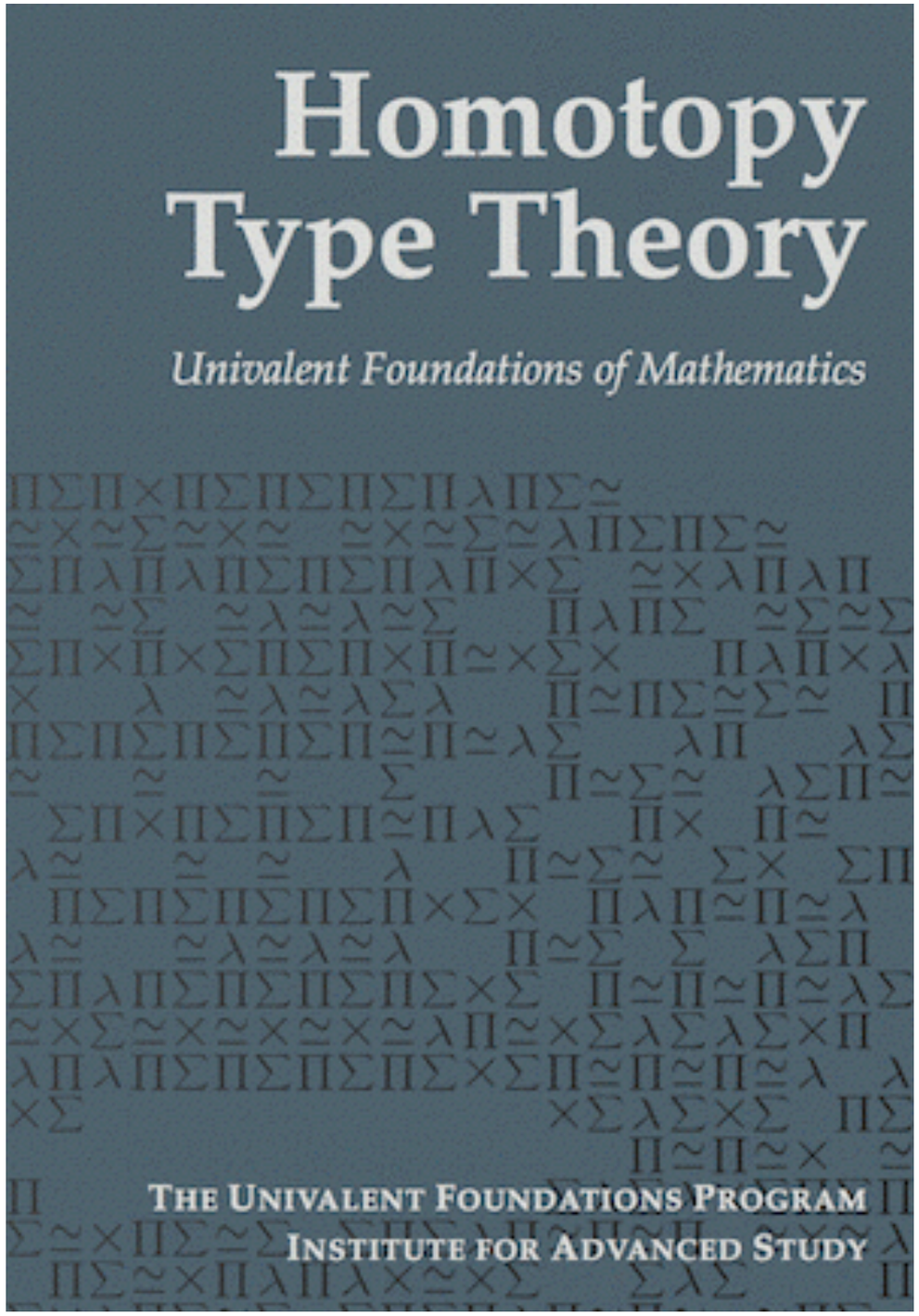
While CIC and other similar theories provide a satisfactory formal environment for the Univalent Foundations we expect to have much better formal environments for it in the future.

The field of mathematics that studies various syntactic and semantic constructions that are required for the representation of mathematics of structures on homotopy types became known as Homotopy Type Theory or HoTT.

An introduction to the most elementary parts of HoTT written from the perspective of comparison with various previous systems of constructive mathematics can be found in the HoTT Book:

It was written by the participants of the Univalent Foundations year at the Institute for Advanced Study in Princeton NJ.

The book is a truly collective effort and as such it does not have an author. The person who did most to make this book happen and who continues to shape the content and the style of the book is Michael Schulman.



Kepler's Conjecture, Four Color Theorem and Feit-Thompson Theorem are all examples of big proofs discovered without the use of proof assistants and later formalized.

Today we are looking forward to the time when proof assistants will become an everyday tool of a working mathematician.

I am already using Coq in my mathematical work and finding ways to define things and to arrange proofs in ways which are convenient for Univalent formalization proves to be useful for making these definitions and proofs more general and better organized.

So far little has been done in univalent formalization of classical mathematics. Most of the work related to the univalent foundations have been concentrated around the exciting and unexpected opportunities which they open up for constructive mathematics.

To test the applicability of Univalent Foundations to actual formalization of mathematics I wrote a library of formalized mathematics for Coq which developed abstract algebra up to the construction of localization of commutative rings.

Approximately at this point the paths of classical and constructive mathematics start to look differently.

Consider for example the notion of a field. A field is defined, classically, as a commutative ring where 0 is not equal to 1 and every non-zero element is invertible.

How to express this condition constructively? We can say one of three things:

- for any x, (x = 0) or (x is invertible)
- for any x, (x is not equal to 0) => (x is invertible)
- for any x, (x is not invertible) => (x = 0)

In constructive logic the first statement implies the the second and the third, but no other implication is available.

And further study of examples such as real numbers shows that in fact it is yet another definition using the notion of apartness which is most appropriate in many cases.

This example show the kind of extra complexity and structure which constructive mathematics has compared to the classical one.

It also shows that classical mathematics provides an important blueprint for the developing of constructive mathematics.

In the next year I plan to work on formalization of classical mathematics in Coq using Univalent Foundations starting with the classical homotopy theory.

This formalization will provide on the one hand a library that research mathematicians in homotopy theory will be able to start using to check their research before or after it was published and on the other an important source for the development of constructive and synthetic homotopy theories.

This formalization of classical homotopy theory will be the first part of a larger project aimed at the formalization and verification of my proof of Milnor's Conjecture.

This formalization will be done in such a way that the researches who work in the related fields will be able to use it to further advance their fields in the familiar classical style or to start exploring the constructive ramifications and structures.