Paul Bernays Lectures, 2014

Prof. Vladimir Voevodsky

from the Institute for Advanced Study, Princeton, NJ, USA

Foundations of Mathematics: their past, present and future.

Part III. Univalent Foundations

In the first lecture we discussed the questions of what is a foundation of mathematics and how foundations of mathematics relate to mathematics.

While trying to answer the first question we observed that there are two streams in the story of foundations. I called these streams "Foundations 1" and 'Foundations 2".

Foundations 1 is defined as the study of "the basic mathematical concepts and how they form hierarchies of more complex structures and concepts".

Foundations 2 is defined as the study of "the fundamentally important structures that form the language of mathematics also called metamathematical concepts".

In the late 19th - early 20th century Foundations 1 went through a momentous transformation. A part of the general theory of forms envisioned by Grassmann was articulated and made precise in the work of Cantor and Dedekind and became known as set theory. New hierarchy of mathematical concepts based on set theory was highly successful and remains in place today.

Foundations 2 went through a period of turbulent development starting with the work of Boole. The discoveries of Kurt Godel freed it from the fruitless search of absolute certainty. Later Foundations 2 became closely associated with the theory and practice of programming languages.

The synthesis of new developments in Foundations 1 and Foundations 2 was attempted in the first decades of the 20th century and led to the development of an interesting and important formal theory called ZFC.

It was later accepted as a postulate that ZFC successfully formalized set theory.

The further development of mathematics proved this postulate to be false. While concepts of set theory such as a set, a subset, a function, and a bijection, took their place as the basis of the new hierarchy of mathematical concepts, ZFC remained known only to specialists in Foundations 2 and played little role in the development of mathematics.

What we have in the Univalent Foundations is the continuation of the development of the "general theory of forms" that Hermann Grassmann spoke about in his 1844 work and that Georg Cantor started to make precise in his theory of sets.

Forms here are being understood in the sense of shapes - this is what homotopy types are, the most fundamental invariants of higher dimensional shapes.

The general "manifolds"or "forms" are called in the Univalent Foundations "types".

The most important new concept of the Univalent Foundations is the concept of h-level.

There is only one, up to the univalent equality, type of h-level 0. It is the one point type.

Types of h-level 1 are propositions. Logic is the study of types of h-level 1.

Types of h-level 2 are sets. The study of such types is the theory of sets and the study of structures on such types is the set-theoretic mathematics.

Types of h-level 3 are the kind of types that are formed by objects in a category. The study of structures on types of h-level 3 is closely related to category-level mathematics.

BTW - there is an important distinction between categorical mathematics and category-level mathematics.

Categorical mathematics is a style of doing set-level mathematics. When working in this style, the class of objects under study is given the structure of a category and the relevant properties of these objects are expressed in terms of the properties of the corresponding objects as objects of an abstract category.

The lower-level analogy of this approach is the study of a class of "things" by defining a partial ordering on the collection of all things of this class and formulating relevant properties of "things" in terms of their position relative to this partial ordering.

Category-level mathematics is the study of structures on categories.

How do we know that we have the right idea of what types of h-level 4 and higher are? After all, we have very little direct experience with such types.

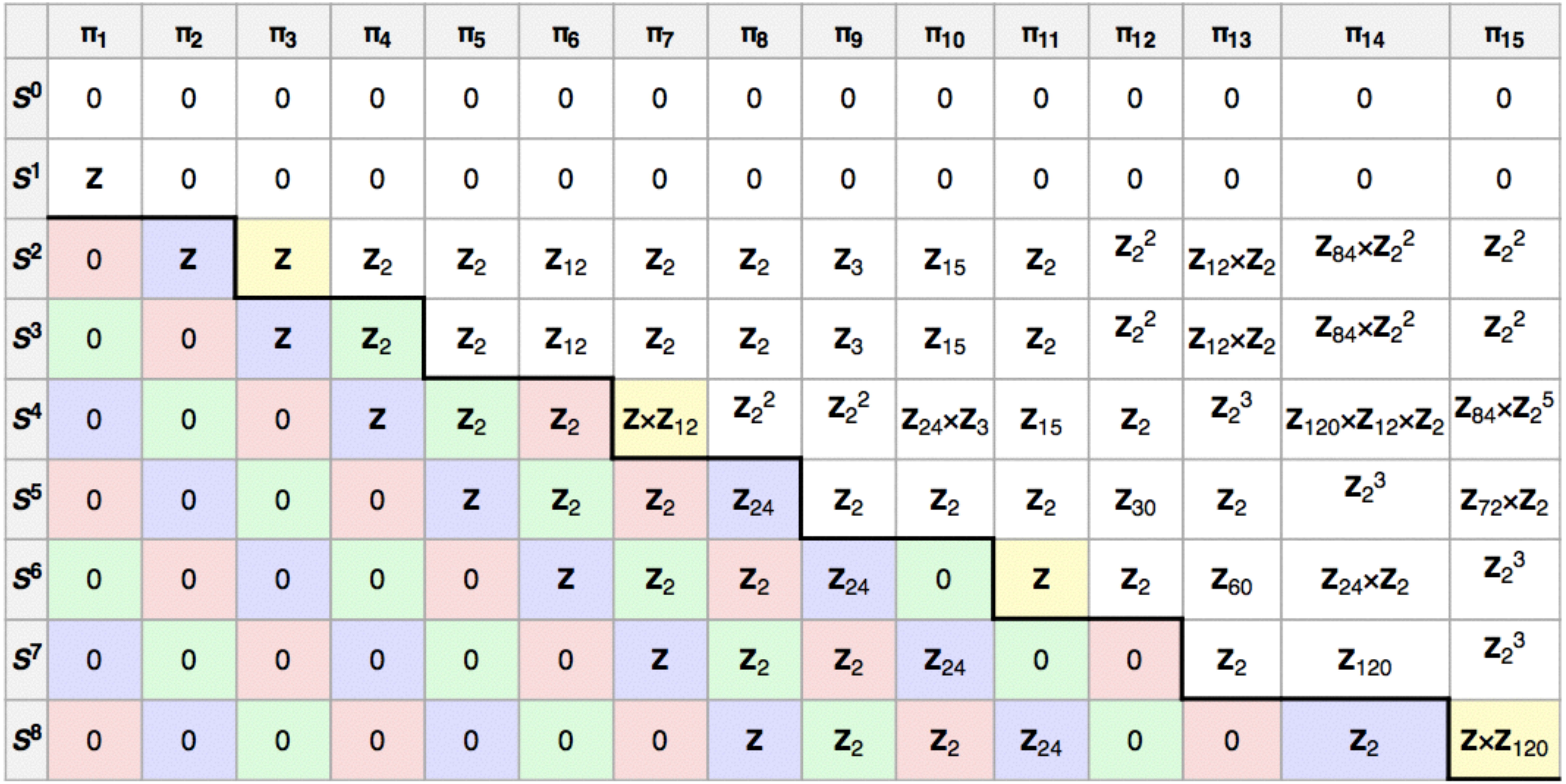
Our certainty comes from the remarkable fact that types of all h-levels appeared in the set-level mathematics in the form of homotopy types.

The theory of homotopy types or homotopy theory (not to be confused with the homotopy type theory!) is a well established field of mathematics that originated with the work of Poincare on the fundamental group.

Homotopy types are equivalence classes of topological spaces with respect to the equivalence relation of being homotopy equivalent.

After their discovery, homotopy types have been appearing in many different areas of modern mathematics. There is a rich and substantial body of knowledge about homotopy types including a large number of highly non-trivial computations.

Here is the table of the first homotopy groups of spheres - an important example of mathematical reality that does not depend on the choice of foundations but requires advanced foundations to be discovered.



	π_1	π_2	π_3	π_4	π_5	π_6	π_7	π_8	π_9	π_{10}	π_{11}	π_{12}	π_{13}	π_{14}	π_{15}
S^0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^1	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S^2	0	Z	Z	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^3	0	0	Z	Z_2	Z_2	Z_{12}	Z_2	Z_2	Z_3	Z_{15}	Z_2	Z_2^2	$Z_{12}\times Z_2$	$Z_{84}\times Z_2^2$	Z_2^2
S^4	0	0	0	Z	Z_2	Z_2	$Z\times Z_{12}$	Z_2^2	Z_2^2	$Z_{24}\times Z_3$	Z_{15}	Z_2	Z_2^3	$Z_{120}\times Z_{12}\times Z_2$	$Z_{84}\times Z_2^5$
S^5	0	0	0	0	Z	Z_2	Z_2	Z_{24}	Z_2	Z_2	Z_2	Z_{30}	Z_2	Z_2^3	$Z_{72}\times Z_2$
S^6	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	Z	Z_2	Z_{60}	$Z_{24}\times Z_2$	Z_2^3
S^7	0	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	0	Z_2	Z_{120}	Z_2^3
S^8	0	0	0	0	0	0	0	Z	Z_2	Z_2	Z_{24}	0	0	Z_2	$Z\times Z_{120}$

From http://en.wikipedia.org/wiki/Homotopy_groups_of_spheres

Univalent Foundations are currently formalized in several styles of Martin-Lof Type Theory.

Martin-Lof introduced his Constructive Type Theory in 1972. It was inspired, in part, by the Automath language created by De Brujin for computer-checkable formalization of mathematics.

The insight that Martin-Lof Type Theory (MLTT) can be used to formalize categorical and higher categorical mathematics belongs to Michael Makkai (see in particular his unpublished paper FOLDS from 1995).

The main mathematical advance that made Univalent Foundations possible is the univalent model of MLTT in the category of simplicial sets that I discovered in 2005 and completed in 2009.

As a part of Univalent Foundations we now have a formalization of set theory in the form of the theory of types of h-level 2 in MLTT.

I believe that this is the first adequate formalization of the set theory that is used in pure mathematics.

Next year I am starting a project of univalent formalization of my proof of Milnor's Conjecture using this formalization of set theory as the starting point.

This project should show that one can conveniently work with classical set-theoretic mathematics in the Univalent Foundations.

It will also provide mathematicians working in the area of homotopy theory and abstract algebraic geometry with the first tools that they can use in their real life work.

We already have a practical formalization of the basic univalent theory of general types, of the basic logic and of the basic set theory. We also have formalization of the basic abstract algebra up to localizations of commutative rings. This is the point where constructive theory starts to substantially diverge from the classical one - there are four interesting constructive concepts that project to the classical concept of a field.

And we have a formalization of the basic category theory.

All these are done with the proof assistant Coq and can be found on GitHub under the name UniMath.

All the formalizations that we have done so far are constructive. For the Milnor's Conjecture Project we are going to allow the use of the Axiom of Excluded Middle (for types of h-level 1) and of the Axiom of Choice (for types of h-level 2) in order to get to interesting areas of pure mathematics sooner.

Let me try to explain now my current understanding of why the theory of sets has not been adequately formalized so far.

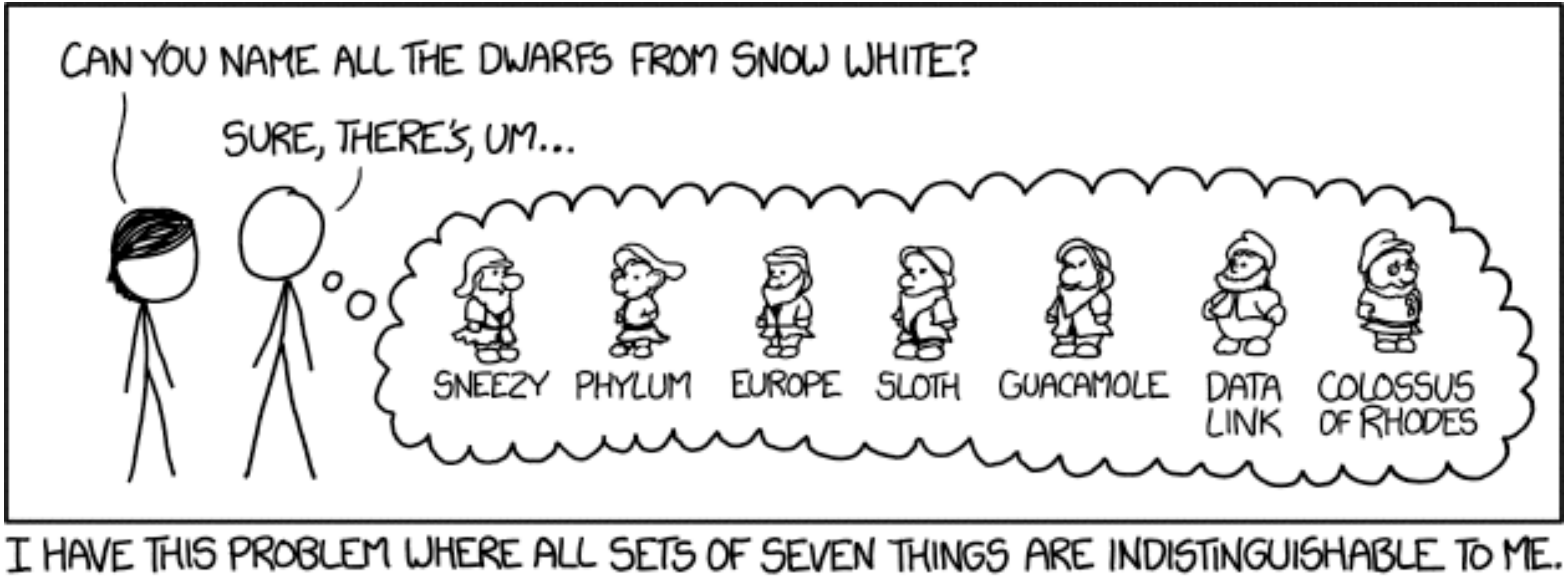
The concept of an abstract set that is used in mathematics is different from the concept of a set of things that is familiar to us from the everyday usage.

The latter presupposes a given universe of things and then by a set with two elements one understands any of the sets of two things from this universe.

The concept of an abstract set does not presupposes any such given universe of things.

An abstract set with two elements refers to an object in the realm of abstract objects that has exactly those properties that are common to all sets of two things and that can be the object of all manipulations that can be applied to all sets of two things.

Clearly any two such abstract objects are in some sense equal to each other. And the same applies to sets with more than two elements.


CAN YOU NAME ALL THE DWARFS FROM SNOW WHITE?
SURE, THERE'S, UM...
SNEEZY
PHYLUM
EUROPE
SLOTH
GUACAMOLE
DATA LINK
COLOSSUS OF RHODES
I HAVE THIS PROBLEM WHERE ALL SETS OF SEVEN THINGS ARE INDISTINGUISHABLE TO ME.

From http://xkcd.com/1417/

More generally, an abstract set corresponding to a given set of things X is an abstract object that has exactly those properties that are common to all sets of things that can be put into a bijective correspondence with X ***and*** that can be the object of all constructions that can be performed on such sets.

But an abstract set is not the class of all sets of things that it is an abstraction of.

An equality between two abstract sets is not a relation but a structure. It is not expressed by a an assertion of a proposition but by presenting an element of a set of possible equalities.

For example, the abstract sets corresponding to the sets {A,B} and {C,D} can be equal either through the correspondence that takes A to C and B to D or through the one that takes A to D and B to C. This is one of the ways in which abstract sets are not “things”.

Logic, at least logic that I am familiar with, can only reason about things and relations between things.

To make logic applicable to abstract sets one has to make "things" from them. This requires a solution to the problem of multiple equalities outlined above. Two such solutions were found.

Georg Cantor developed a theory of well-orderings on sets. Choosing a well-ordering turns a set into a "thing" because two sets with well-orderings can be equal to each other in no more than one way. Cantor called these "things" transfinite numbers.

Objects considered by Zermelo can be defined inductively saying that there is a distinguished Zermelo object called Φ and that any non-empty collection of pairwise different Zermelo objects is a Zermelo object.

Every Zermelo object defines a set. Φ is said to define the empty set, while a composite Zermelo object defines the set of the Zermelo objects from which it is built.

We can also say that a Zermelo object defines a set with a Zermelo structure, where a Zermelo structure on a set is an assignment of Zermelo objects to elements of the set in such a way that different elements are assigned different Zermelo objects.

Since Zermelo objects from which a composite Zermelo object is built are all different two Zermelo objects can be equal to each other in no more that one way - Zermelo objects are "things".

Zermelo objects are more versatile than well-ordered sets and one can define the main constructions that one need to perform on sets on the level of Zermelo objects.

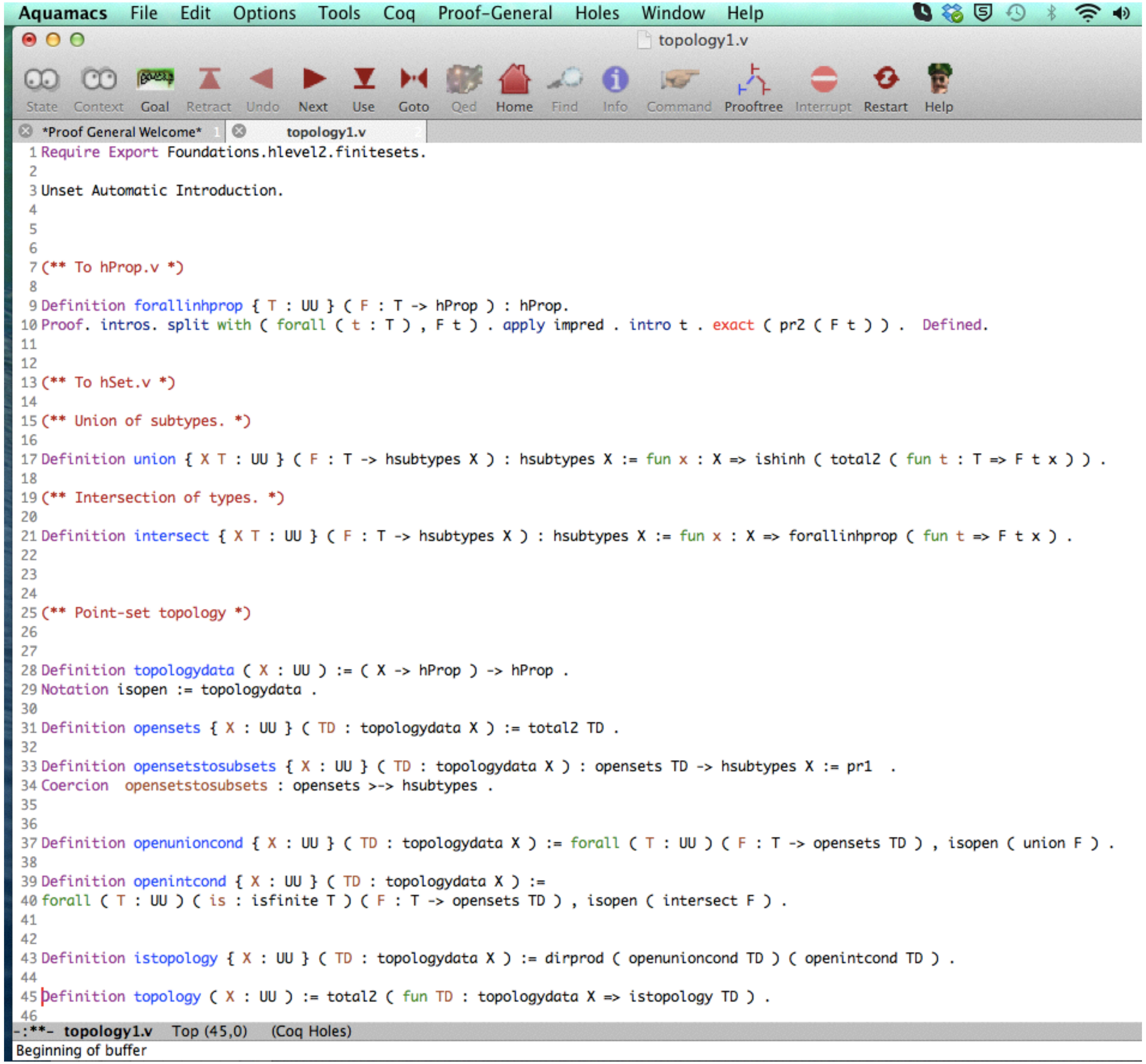
This almost made it possible to hide the fact that Zermelo objects are not the same as sets. The difference became apparent in two ways.

Firstly, relations could be defined between Zermelo objects that made no sense in the language of sets. For example, one could always ask whether Φ is an element of a given Zermelo object X. This can not be asked of an abstract set corresponding to X because some of the sets in a bijective correspondence with X will have Φ as an element and some won't.

Secondly, anyone using Zermelo objects to represent sets was forced to make a lot of arbitrary choices. For example, the number 1 encoded in set theory as the set with one element could be encoded by many, non equal, Zermelo objects.

Here is a screenshot of an actual development in the proof assistant Coq based on the univalent foundations.

This page provides the definition of a topological space together with some technical constructions that are needed for this definition such as the constructions of the union and intersection of a family of sub-types.



```
Require Export Foundations.hlevel2.finitesets.

Unset Automatic Introduction.



(** To hProp.v *)

Definition forallinhprop { T : UU } ( F : T -> hProp ) : hProp.
Proof. intros. split with ( forall ( t : T ) , F t ) . apply impred . intro t . exact ( pr2 ( F t ) ) .  Defined.


(** To hSet.v *)

(** Union of subtypes. *)

Definition union { X T : UU } ( F : T -> hsubtypes X ) : hsubtypes X := fun x : X => ishinh ( total2 ( fun t : T => F t x ) ) .

(** Intersection of types. *)

Definition intersect { X T : UU } ( F : T -> hsubtypes X ) : hsubtypes X := fun x : X => forallinhprop ( fun t => F t x ) .



(** Point-set topology *)


Definition topologydata ( X : UU ) := ( X -> hProp ) -> hProp .
Notation isopen := topologydata .

Definition opensets { X : UU } ( TD : topologydata X ) := total2 TD .

Definition opensetstosubsets { X : UU } ( TD : topologydata X ) : opensets TD -> hsubtypes X := pr1  .
Coercion  opensetstosubsets : opensets >-> hsubtypes .


Definition openunioncond { X : UU } ( TD : topologydata X ) := forall ( T : UU ) ( F : T -> opensets TD ) , isopen ( union F ) .

Definition openintcond { X : UU } ( TD : topologydata X ) :=
forall ( T : UU ) ( is : isfinite T ) ( F : T -> opensets TD ) , isopen ( intersect F ) .


Definition istopology { X : UU } ( TD : topologydata X ) := dirprod ( openunioncond TD ) ( openintcond TD ) .

Definition topology ( X : UU ) := total2 ( fun TD : topologydata X => istopology TD ) .

```

```
-:**-  topology1.v    Top (45,0)    (Coq Holes)
Beginning of buffer
```

In the rest of this lecture I want to discuss some of the issues that I consider important for foundations of mathematics and mathematics as a whole.

In the Univalent Foundations we accept Godel's results not simply as correct ones but as ***natural*** ones.

We know that any foundation will be incomplete and we are not concerned with the impossible task of finding one complete foundation.

Instead, we are concerned with creating a practical foundation which we can use now and establishing a process which can ensure a healthy grows and transformation of this foundation in the future.

We are close to having a practical foundation that we can use now. Work on such a foundation continues with the two main issues at hand being the implementations of a strong substitutional equality and efficient universe management.

The computer software used in the process of association of formal objects with thoughts and of verification of the formal properties of these objects is becoming generally known as **computer proof assistants**.

There are many proof assistants in use and development today. The most interesting thing happening in connection with the development of proof assistants are the ongoing **changes** in what methods of reasoning are allowed **at the thought level** in the framework of rigorous mathematical arguments.

Prior to the appearance of the univalent foundations the main innovations in this area were related to:

1. The use of complex dependencies of types.

2. The use of complex inductive constructions way beyond the complexity allowed by traditional foundations.

The univalent foundations brought yet another aspect to these changes - the ability to reason about structures on the higher analogs of sets using **Univalence Axiom.** The work being done in particular, by Thierry Coquand and his colleagues leads to concrete algorithms which show that the Univalence Axiom can be used not only consistently but also constructively.

I expect that this process of addition of new sophisticated methods of reasoning at the thought level, made possible by the ability of computers to ensure that these methods are used correctly, will continue in the coming years.

There is one important thing that we can and should do to make this great process of innovation to advance smoothly. We need to make sure that the consistency of these new methods of reasoning is grounded in the consistency of the known methods of reasoning through the use of classical formal methods aided by computers.

Explicitly speaking we should:

Create the infrastructure for formalization, in the framework of established formal deduction systems, of the mathematical meta-theory of new formal deduction systems that can be used to connect the newly emerging sophisticated methods of reasoning to the systems of reasoning whose consistency has been established through many years, decades and centuries of use.

In practical terms, we should be looking for the system when a newly invented reasoning paradigm, e.g. a new class of higher induction-recursion, described formally as a collection of syntax-based rules for the construction of new classes of sentences and inferences with these sentences, can undergo **a formal consistency certification process**.

In that process, possibly through the construction of sophisticated models, certain combinations of the new rules with the existing ones will be certified as being equi-consistent with one of the etalons of consistency such as, for example, ZFC.

After that there remains the issue of faithful implementation of these rules in a proof assistant but that issue is of a lesser order of complexity.

Such a system will give us a real freedom to experiment with new methods of reasoning relying only on our intuition of their consistency because we will know that our intuition is based on previous experience which has been certified to be free of errors and because we will know that it is safe to make mistakes since these mistakes will be localized on later stages of development and options for their correction, without the need to re-work all that has been done, will be provided.