FOUNDATIONS OF MATHEMATICS

Will Computers Redefine the Roots of Math?

By KEVIN HARTNETT

May 19, 2015

When a legendary mathematician found a mistake in his own work, he embarked on a computer-aided quest to eliminate human error. To succeed, he has to rewrite the century-old rules underlying all of mathematics.

34 |

Mathematics is being rebuilt on top of new foundations.

—

Hannes Hummel for Quanta Magazine

On a recent train trip from Lyon to Paris, Vladimir Voevodsky sat next to Steve Awodey and tried to convince him to change the way he does mathematics.

Voevodsky, 48, is a permanent faculty member at the Institute for Advanced Study (IAS) in Princeton, N.J. He was born in Moscow but speaks nearly flawless English, and he has the confident bearing of someone who has no need to prove himself to anyone. In 2002 he won the Fields Medal, which is often considered the most prestigious award in mathematics.

Now, as their train approached the city, Voevodsky pulled out his laptop and opened a program called Coq, a proof assistant that provides mathematicians with an environment in which to write mathematical arguments. Awodey, a mathematician and logician at Carnegie Mellon University in Pittsburgh, Pa., followed along as Voevodsky wrote a definition of a mathematical object using a new formalism he had created, called univalent foundations. It took Voevodsky 15 minutes to write the definition.

"I was trying to convince [Awodey] to do [his mathematics in Coq]," Voevodsky explained during a lecture this past fall. "I was trying to convince him that it's easy to do."

The idea of doing mathematics in a program like Coq has a long history. The appeal is simple: Rather than relying on fallible human beings to check proofs, you can turn the job over to computers, which can tell whether a proof is correct with complete certainty. Despite this advantage, computer proof assistants haven't been widely adopted in mainstream mathematics. This is partly because translating everyday math into terms a computer can understand is cumbersome and, in the eyes of many mathematicians, not worth the effort.

For nearly a decade, Voevodsky has been advocating the virtues of computer proof assistants and developing univalent foundations in order to bring the languages of mathematics and computer programming closer together. As he sees it, the move to computer formalization is necessary because some branches of mathematics have become too abstract to be reliably checked by people.

"The world of mathematics is becoming very large, the complexity of mathematics is becoming very high, and there is a danger of an accumulation of mistakes," Voevodsky said. Proofs rely on other proofs; if one contains a flaw, all others that rely on it will share the error.

This is something Voevodsky has learned through personal experience. In 1999 he discovered an error in a paper he had written seven years earlier. Voevodsky eventually found a way to salvage the result, but in an article last summer in the IAS newsletter, he wrote that the experience scared him. He began to worry that unless he formalized his work on the computer, he wouldn't have complete confidence that it was correct.

But taking that step required him to rethink the very basics of mathematics. The accepted foundation of mathematics is set theory. Like any foundational system, set theory provides a collection of basic concepts and rules, which can be used to construct the rest of mathematics. Set theory has sufficed as a foundation for more than a century, but it can't readily be translated into a form that computers can use to check proofs. So with his decision to start formalizing mathematics on the computer, Voevodsky set in motion a process of discovery that ultimately led to something far more ambitious: a recasting of the underpinnings of mathematics.

Set Theory and a Paradox

Set theory grew out of an impulse to put mathematics on an entirely rigorous footing — a logical basis even more secure than numbers themselves. Set theory begins with the set containing nothing — the null set — which is used to define the number zero. The number 1 can then be built by defining a new set with one element — the null set. The number 2 is the set that contains two elements — the null set (0) and the set that contains the null set (1). In this way, each whole number can be defined as the set of sets that came before it.

Set theory constructs all of mathematics by starting with literally nothing — the null set — which is defined to be zero. The set that contains a single element — the null set — becomes the number 1, the set that contains two elements — the null set and the set containing the null set — becomes the number 2, and so on.

Olena Shmahalo/Quanta Magazine

Once the whole numbers are in place, fractions can be defined as pairs of whole numbers, decimals can be defined as sequences of digits, functions in the plane can be defined as sets of ordered pairs, and so on. "You end up with complicated structures, which are a set of things, which are a set of things, which are a set of things, all the way down to the metal, to the empty set at the bottom," said Michael Shulman, a mathematician at the University of San Diego.

Set theory as a foundation includes these basic objects — sets — and logical rules for manipulating those sets, from which the theorems in mathematics are derived. An advantage of set theory as a foundational system is that it is very economical — every object mathematicians could possibly want to use is ultimately built from the null set.

On the other hand, it can be tedious to encode complicated mathematical objects as elaborate hierarchies of sets. This limitation becomes problematic when mathematicians want to think about objects that are equivalent or isomorphic in some sense, if not necessarily equal in all respects. For example, the fraction ½ and the decimal 0.5 represent the same real number but are encoded very differently in terms of sets.

"You have to build up a specific object and you're stuck with it," Awodey said. "If you want to work with a different but isomorphic object, you'd have to build that up."

But set theory isn't the only way to do mathematics. The proof assistant programs Coq and Agda, for example, are based on a different formal system called type theory.

Type theory has its origins in an attempt to fix a critical flaw in early versions of set theory, which was identified by the philosopher and logician Bertrand Russell in 1901. Russell noted that some sets contain themselves as a member. For example, consider the set of all things that are not spaceships. This set — the set of non-spaceships — is itself not a spaceship, so it is a member of itself.

Russell defined a new set: the set of all sets that do not contain themselves. He asked whether that set contains itself, and he showed that answering that question produces a paradox: If the set does contain itself, then it doesn't contain itself (because the only objects in the set are sets that don't contain themselves). But if it doesn't contain itself, it does contain itself (because the set contains all the sets that don't contain themselves).

Russell created type theory as a way out of this paradox. In place of set theory, Russell's system used more carefully defined objects called types. Russell's type theory begins with a universe of objects, just like set theory, and those objects can be collected in a "type" called a *SET*. Within type theory, the type *SET* is defined so that it is only allowed to collect objects that aren't collections of other things. If a collection does contain other collections, it is no longer allowed to be a *SET*, but is instead something that can be thought of as a *MEGASET* — a new kind of type defined specifically as a collection of objects which themselves are collections of objects.

From here, the whole system arises in an orderly fashion. One can imagine, say, a type called a *SUPERMEGASET* that collects only objects that are *MEGASETS*. Within this rigid framework, it becomes illegal, so to speak, to even ask the paradox-inducing question, "Does the set of all sets that do not contain themselves contain itself?" In type theory, *SETS* only contain objects that are not collections of other objects.

An important distinction between set theory and type theory lies in the way theorems are treated. In set theory, a theorem is not itself a set — it's a statement about sets. By contrast, in some versions of type theory, theorems and *SETS* are on equal footing. They are "types" — a new kind of mathematical object. A theorem is the type whose elements are all the different ways the theorem can be proved. So, for example, there is a single type that collects all the proofs to the Pythagorean theorem.

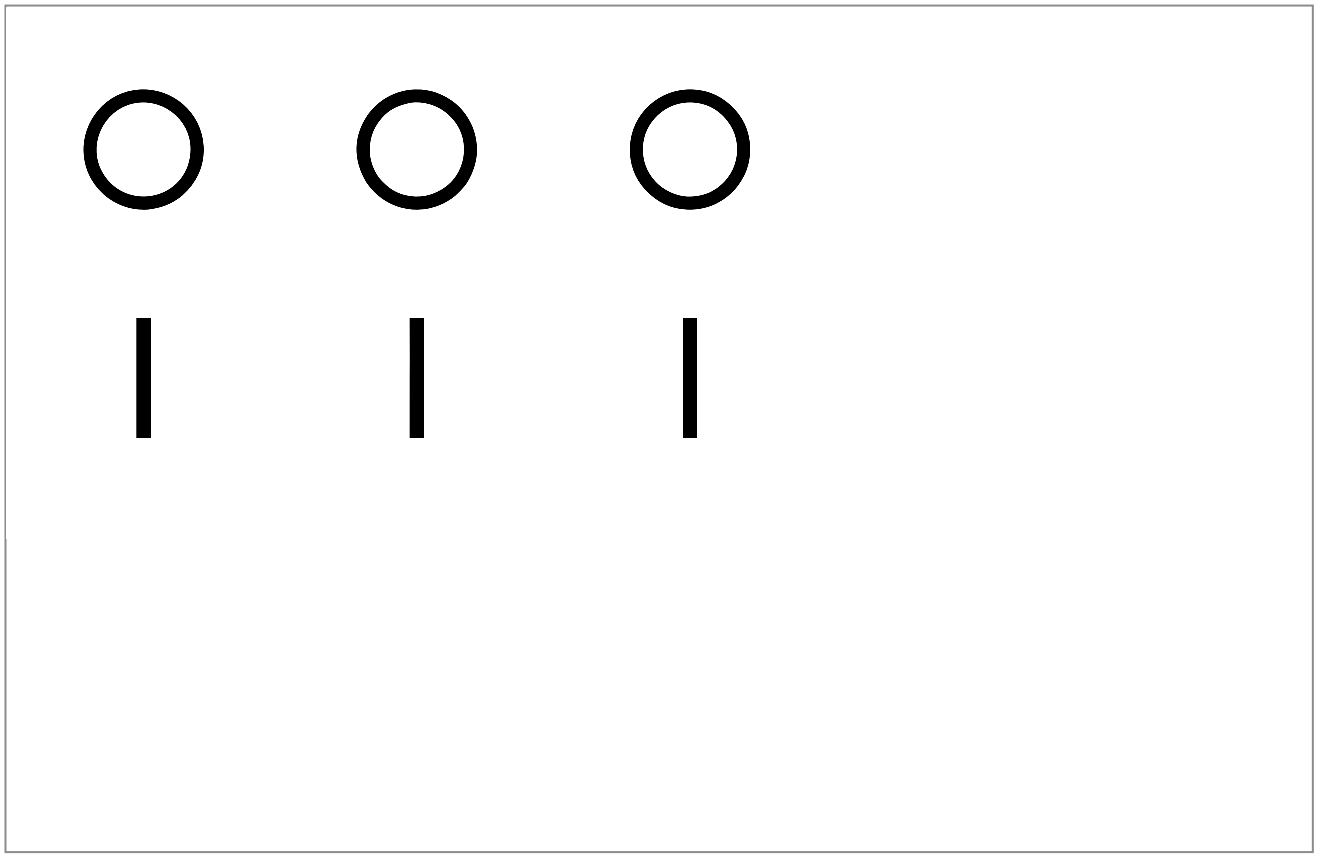
To illustrate this difference between set theory and type theory, consider two sets: Set *A* contains two apples and Set *B* contains two oranges. A mathematician might consider these sets equivalent, or isomorphic, because they have the same number of objects. The way to show formally that these two sets are equivalent is to pair objects from the first set with objects from the second. If they pair evenly, with no objects left over on either side, they're equivalent.

When you do this pairing, you quickly see that there are two ways to show the equivalence: Apple 1 can be paired with Orange 1 and Apple 2 with Orange 2, or Apple 1 can be paired with Orange 2 and Apple 2 with Orange 1. Another way to say this is to state that the two sets are isomorphic to each other in two ways.

In a traditional set theory proof of the theorem Set $A \cong$ Set B (where the symbol $\cong$ means "is isomorphic to"), mathematicians are concerned only with whether one such pairing exists. In type theory, the theorem Set $A \cong$ Set B can be interpreted as a collection, consisting of all the different ways of demonstrating the isomorphism (which in this case is two). There are often good reasons in mathematics to keep track of the various ways in which two objects (like these two sets) are equivalent, and type theory does this automatically by bundling equivalences together into a single type.

This is especially useful in topology, a branch of mathematics that studies the intrinsic properties of spaces, like a circle or the surface of a doughnut. Studying spaces would be impractical if topologists had to think separately about all possible variations of spaces with the same intrinsic properties. (For example, circles can come in any size, yet every circle shares the same basic qualities.) A solution is to reduce the number of distinct spaces by considering some of them to be equivalent. One way topologists do this is with the notion of homotopy, which provides a useful definition of equivalence: Spaces are homotopy equivalent if, roughly speaking, one can be deformed into the other by shrinking or thickening regions, without tearing.

The point and the line are homotopy equivalent, which is another way of saying they're of the same homotopy type. The letter *P* is of the same homotopy type as the letter *O* (the tail of the *P* can be collapsed to a point on the boundary of the letter's upper circle), and both *P* and *O* are of the same homotopy type as the other letters of the alphabet that contain one hole — *A*, *D*, *Q* and *R*.



Emily Fuhrman for Quanta Magazine

Homotopy types are used to classify the essential features of an object. The letters A, R and Q all have one hole, and so are the same homotopy type. The letters C, X and K are also of the same homotopy type, as they can all transform into a line.

Topologists use different methods for assessing the qualities of a space and determining its homotopy type. One way is to study the collection of paths between distinct points in the space, and type theory is well-suited to keeping track of these paths. For instance, a topologist might think of two points in a space as equivalent whenever there is a path connecting them. Then the collection of all paths between points *x* and *y* can itself be viewed as a single type, which represents all proofs of the theorem *x* = *y*.

Homotopy types can be constructed from paths between points, but an enterprising mathematician can also keep track of paths between paths, and paths between paths between paths, and so on. These paths between paths can be thought of as higher-order relationships between points in a space.

Voevodsky tried on and off for 20 years, starting as an undergraduate at Moscow State University in the mid-1980s, to formalize mathematics in a way that would make these higher-order relationships — paths of paths of paths — easy to work with. Like many others during this period, he tried to accomplish this within the framework of a formal system called category theory. And while he achieved limited success in using category theory to formalize particular regions of mathematics, there remained regions of mathematics that categories couldn't reach.

Voevodsky returned to the problem of studying higher-order relationships with renewed interest in the years after he won the Fields Medal. In late 2005, he had something of an epiphany. As soon as he started thinking about higher-order relationships in terms of objects called infinity-groupoids, he said, "many things started to fall into place."

Infinity-groupoids encode all the paths in a space, including paths of paths, and paths of paths of paths. They crop up in other frontiers of mathematical research as ways of encoding similar higher-order relationships, but they are unwieldy objects from the point of view of set theory. Because of this, they were thought to be useless for Voevodsky's goal of formalizing mathematics.

Yet Voevodsky was able to create an interpretation of type theory in the language of infinity-groupoids, an advance that allows mathematicians to reason efficiently about infinity-groupoids without ever having to think of them in terms of sets. This advance ultimately led to the development of univalent foundations.

Voevodsky was excited by the potential of a formal system built on groupoids, but also daunted by the amount of technical work required to realize the idea. He was also concerned that any progress he made would be too complicated to be reliably verified through peer review, which Voevodsky said he was "losing faith in" at the time.

Toward a New Foundational System

With groupoids, Voevodsky had his object, which left him needing only a formal framework in which to organize them. In 2005 he found it in an unpublished paper called FOLDS, which introduced Voevodsky to a formal system that fit uncannily well with the kind of higher-order mathematics he wanted to practice.

In 1972 the Swedish logician Per Martin-Löf introduced his own version of type theory inspired by ideas from Automath, a formal language for checking proofs on the computer. Martin-Löf type theory (MLTT) was eagerly adopted by computer scientists, who have used it as the basis for proof-assistant programs.

In the mid-1990s, MLTT intersected with pure mathematics when Michael Makkai, a specialist in mathematical logic who retired from McGill University in 2010, realized it might be used to formalize categorical and higher-categorical mathematics. Voevodsky said that when he first read Makkai's work, set forth in FOLDS, the experience was "almost like talking to myself, in a good sense."

Vladimir Voevodsky's univalent foundations program aims to rebuild mathematics in a way that will allow computers to check all mathematical proofs.

Andrea Kane/Institute for Advanced Study

Voevodsky followed Makkai's path but used groupoids instead of categories. This allowed him to create deep connections between homotopy theory and type theory.

"This is one of the most magical things, that somehow it happened that these programmers really wanted to formalize [type theory]," Shulman said, "and it turns out they ended up formalizing homotopy theory."

Voevodsky agrees that the connection is magical, though he sees the significance a little differently. To him, the real potential of type theory informed by homotopy theory is as a new foundation for mathematics that's uniquely well-suited both to computerized verification and to studying higher-order relationships.

Voevodsky first perceived this connection when he read Makkai's paper, but it took him another four years to make it mathematically precise. From 2005 to 2009 Voevodsky developed several pieces of machinery that allow mathematicians to work with sets in MLTT "in a consistent and convenient way for the first time," he said. These include a new axiom, known as the univalence axiom, and a complete interpretation of MLTT in the language of simplicial sets, which (in addition to groupoids) are another way of representing homotopy types.

This consistency and convenience reflects something deeper about the program, said Daniel Grayson, an emeritus professor of mathematics at the University of Illinois at Urbana-Champaign. The strength of univalent foundations lies in the fact that it taps into a previously hidden structure in mathematics.

"What's appealing and different about [univalent foundations], especially if you start viewing [it] as replacing set theory," he said, "is that it appears that ideas from topology come into the very foundation of mathematics."

From Idea to Action

Building a new foundation for mathematics is one thing. Getting people to use it is another. By late 2009 Voevodsky had worked out the details of univalent foundations and felt ready to begin sharing his ideas. He understood people were likely to be skeptical. "It's a big thing to say I have something which should probably replace set theory," he said.

Voevodsky first discussed univalent foundations publicly in lectures at Carnegie Mellon in early 2010 and at the Oberwolfach Research Institute for Mathematics in Germany in 2011. At the Carnegie Mellon talks he met Steve Awodey, who had been doing research with his graduate students Michael Warren and Peter Lumsdaine on homotopy type theory. Soon after, Voevodsky decided to bring researchers together for a period of intensive collaboration in order to jump-start the field's development.

Along with Thierry Coquand, a computer scientist at the University of Gothenburg in Sweden, Voevodsky and Awodey organized a special research year to take place at IAS during the 2012-2013 academic year. More than thirty computer scientists, logicians and mathematicians came from around the world to participate. Voevodsky said the ideas they discussed were so strange that at the outset, "there wasn't a single person there who was entirely comfortable with it."

The ideas may have been slightly alien, but they were also exciting. Shulman deferred the start of a new job in order to take part in the project. "I think a lot of us felt we were on the cusp of something big, something really important," he said, "and it was worth making some sacrifices to be involved with the genesis of it."

Following the special research year, activity split in a few different directions. One group of researchers, which includes Shulman and is referred to as the HoTT community (for homotopy type theory), set off to explore the possibilities for new discoveries within the framework they'd developed. Another group, which identifies as UniMath and includes Voevodsky, began rewriting mathematics in the language of univalent foundations. Their goal is to create a library of basic mathematical elements — lemmas, proofs, propositions — that mathematicians can use to formalize their own work in univalent foundations.

As the HoTT and UniMath communities have grown, the ideas that underlie them have become more visible among mathematicians, logicians and computer scientists. Henry Towsner, a logician at the University of Pennsylvania, said that there seems to be at least one presentation on homotopy type theory at every conference he attends these days, and that the more he learns about the approach, the more it makes sense. "It was this buzzword," he said. "It took me awhile to understand what they were actually doing and why it was interesting and a good idea, not a gimmicky thing."

A lot of the attention univalent foundations has received is owing to Voevodsky's standing as one of the greatest mathematicians of his generation. Michael Harris, a mathematician at Columbia University, includes a long discussion of univalent foundations in his new book, *Mathematics Without Apologies*. He is impressed by the mathematics that surround the univalence model, but he is more skeptical of Voevodsky's larger vision of a world in which all mathematicians formalize their work in univalent foundations and check it on the computer.

"The drive to mechanize proof and proof verification doesn't strongly motivate most mathematicians as far as I can tell," he said. "I can understand why computer scientists and logicians would be excited, but I think mathematicians are looking for something else."

Voevodsky is aware that a new foundation for mathematics is a tough sell, and he admits that "at the moment there is really more hype and noise than the field is ready for." He is currently using the language of univalent foundations to formalize the relationship between MLTT and homotopy theory, which he considers a necessary next step in the development of the field. Voevodsky also has plans to formalize his proof of the Milnor conjecture, the achievement for which he earned a Fields Medal. He hopes that doing so might act as "a milestone which can be used to create motivation in the field."

Voevodsky would eventually like to use univalent foundations to study aspects of mathematics that have been inaccessible within the framework of set theory. But for now he's approaching the development of univalent foundations cautiously. Set theory has undergirded mathematics for more than a century, and if univalent foundations is to have similar longevity, Voevodsky knows it's important to get things right at the start.

This article was reprinted on Wired.com.