Univalent Foundations - new type-theoretic foundations of mathematics.

Talk at IHP, Paris on April 22 , 2014

by Vladimir Voevodsky

from Institute for Advanced Study in Princeton, NJ.

First, let me state the following over simplified, but, I believe, important for the understanding thesis:

The main idea of the Univalent Foundations is how to interpret type-theoretic universes mathematically.

In answering this question I was strongly influenced by the paper of M. Makkai "First Order Logic with Dependent Sorts, with Applications to Category Theory”. There Makkai discusses what he sees as the future foundations of mathematics and even gives them a name: “Invariant Foundations”.

He then writes:

”The universe of the Invariant Foundation is not clearly defined as yet. It should contain ana-n-categories for all natural n’s; the totality of ana-n-categories, with their morphisms, etc., will form an ana-n+1-category.”

It is a very natural idea and it took me a lot of effort to understand that it is *wrong* and that the universe of the new foundations of mathematics should not be the ∞-category of ∞-categories but instead the ∞-groupoid of ∞-groupoids and their equivalences.

Unlike ∞-categories, which we still do not have a good understanding of, the ∞-groupoids are easy to understand due to the reversal of an idea of A. Grothendieck which he expresses as follows:

… the intuition appeared that ∞-groupoids should constitute particularly adequate models for homotopy types, the n-groupoids corresponding to truncated homotopy types (with π_i = 0 pour i > n).

A. Grothendieck "Esquisse d'un Programme” 1984.

According to this “Grothendieck correspondence”, which M. Kapranov and I provided a wrong formulation of but which remains true with correct definitions, ∞-groupoids are models for homotopy types.

Hence, there should be a homotopy type corresponding to the ∞-groupoid of ∞-groupoids and their equivalences.

What is it?

Note that the previous line of reasoning is done in an inconsistent reasoning system which assumes that there is such a thing as an ∞-groupoid of *all* ∞-groupoids.

In a more complex system of ZFC, which we believe to be consistent, the reasoning becomes more complex. Instead of talking about the ∞-groupoid of all ∞-groupoids we should fix two set-theoretic universes U0 and U1, such that U0 is an element of U1, and talk about ∞-groupoids and their equivalences in U0 as an ∞-groupoid in U1.

This ∞-groupoid should correspond to a well-defined homotopy type U=U(U0) in U1.

Can we construct this homotopy type directly?

The main thing we know about U is that there should be a fibration U’ -> U over U which corresponds to the fibration over the ∞-groupoid of ∞-groupoids whose fiber over an object is the ∞-groupoid representing this object.

In the simplified reasoning system where there is a set of all sets this would be the universal fibration i.e. the fibration which classifies all fibrations.

In the more complex reasoning system with universes this fibration should still satisfy the “uniqueness” part of the definition of “universal” but not the “for all” part.

Fibrations which satisfy the “for all” part but not the uniqueness part are known in algebraic geometry as “versal fibrations”.

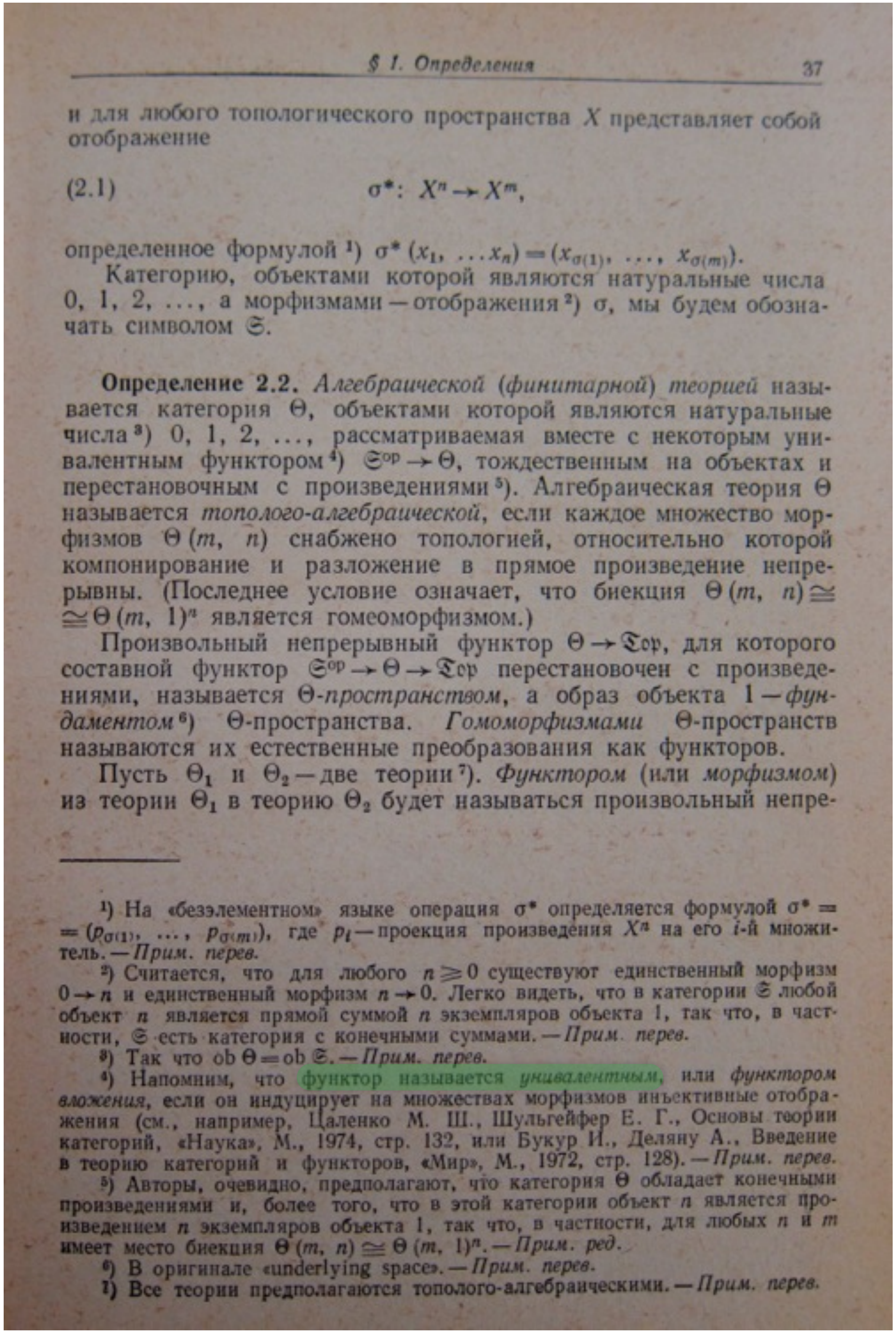
I was looking for a word to use for fibrations which satisfy the uniqueness part but not the “for all” part and decided to call them “univalent”. And this is why the foundations are called Univalent Foundations.

But there is a little more to the story of this name.

When I decided to check something in the Russian translation of the Boardman and Vogt book “*Homotopy Invariant Algebraic Structures on Topological Spaces*” I discovered that in this book the term “faithful functor” was translated as ‘univalent functor” (“унивалентный функтор”).

Since I have tried to read this book in my youth many times there was probably another meaning associated in my mind with the word “univalent” - “faithful”.

Indeed these foundations seem to be faithful to the way in which I think about mathematical objects in my head.



§ *1. Определения* 37

и для любого топологического пространства X представляет собой отображение

$$(2.1) \qquad \sigma^*\colon X^n \to X^m,$$

определенное формулой [1] $\sigma^*(x_1, \ldots x_n) = (x_{\sigma(1)}, \ldots, x_{\sigma(m)})$.

Категорию, объектами которой являются натуральные числа 0, 1, 2, ..., а морфизмами — отображения [2] σ, мы будем обозначать символом $\mathfrak{S}$.

Определение 2.2. *Алгебраической* (*финитарной*) *теорией* называется категория Θ, объектами которой являются натуральные числа [3] 0, 1, 2, ..., рассматриваемая вместе с некоторым унивалентным функтором [4] $\mathfrak{S}^{\mathrm{op}} \to \Theta$, тождественным на объектах и перестановочным с произведениями [5]. Алгебраическая теория Θ называется *тополого-алгебраической*, если каждое множество морфизмов $\Theta(m, n)$ снабжено топологией, относительно которой компонирование и разложение в прямое произведение непрерывны. (Последнее условие означает, что биекция $\Theta(m, n) \cong \Theta(m, 1)^n$ является гомеоморфизмом.)

Произвольный непрерывный функтор $\Theta \to \mathfrak{Top}$, для которого составной функтор $\mathfrak{S}^{\mathrm{op}} \to \Theta \to \mathfrak{Top}$ перестановочен с произведениями, называется Θ-*пространством*, а образ объекта 1 — *фундаментом* [6] Θ-пространства. *Гомоморфизмами* Θ-пространств называются их естественные преобразования как функторов.

Пусть Θ_1 и Θ_2 — две теории [7]. *Функтором* (или *морфизмом*) из теории Θ_1 в теорию Θ_2 будет называться произвольный непре-

[1] На «безэлементном» языке операция σ^* определяется формулой $\sigma^* = (p_{\sigma(1)}, \ldots, p_{\sigma(m)})$, где p_i — проекция произведения X^n на его i-й множитель. — *Прим. перев.*

[2] Считается, что для любого $n \geqslant 0$ существуют единственный морфизм $0 \to n$ и единственный морфизм $n \to 0$. Легко видеть, что в категории $\mathfrak{S}$ любой объект n является прямой суммой n экземпляров объекта 1, так что, в частности, $\mathfrak{S}$ есть категория с конечными суммами. — *Прим. перев.*

[3] Так что $\mathrm{ob}\,\Theta = \mathrm{ob}\,\mathfrak{S}$. — *Прим. перев.*

[4] Напомним, что функтор называется *унивалентным*, или *функтором вложения*, если он индуцирует на множествах морфизмов инъективные отображения (см., например, Цаленко М. Ш., Шульгейфер Е. Г., Основы теории категорий, «Наука», М., 1974, стр. 132, или Букур И., Деляну А., Введение в теорию категорий и функторов, «Мир», М., 1972, стр. 128). — *Прим. перев.*

[5] Авторы, очевидно, предполагают, что категория Θ обладает конечными произведениями и, более того, что в этой категории объект n является произведением n экземпляров объекта 1, так что, в частности, для любых n и m имеет место биекция $\Theta(m, n) \cong \Theta(m, 1)^n$. — *Прим. ред.*

[6] В оригинале «underlying space». — *Прим. перев.*

[7] Все теории предполагаются тополого-алгебраическими. — *Прим. перев.*

The main ideas of Univalent Foundations have now been used in at least two libraries of formalized mathematics in Coq - the “HoTT library” and the “UniMath library”.

There have also been interesting proofs using the univalent ideas done in Agda.

But it appears that further progress in the direction of developing of proof assistants for the everyday use by mathematicians requires us to find ways to collaborate with the communities using other proof assistant architectures.

One such collaboration may be approaching reality now thanks in a large part to the help of Bill Richter - a homotopy theorist turned HOL Light programmer.

His mediation is supporting an interaction which has recently started between me and John Harrison and opened up for me the ideas of the LCF-architecture.

The slides in the rest of my talk are more technical and only partially complete. They are just pointers to the ideas which I think require further attention.

What does one need in a type-theoretic proof system in order to express the main ideas of Univalent Foundations?

1. A universe U.

2. Dependent products.

3. Dependent sums.

Do we need a sequence of embedded universes?

Yes - if we a looking for a consistent system,

No - if we do not care about formal consistency.

Let me explain why the second perspective is very important. We want to have the ideas of univalence to be useful not only in mathematics but also in computer science.

This means designing “univalent programing languages”.

And most successful programming languages are inconsistent deduction systems - it is possible to write non-terminating programs in these languages.

Building a type system where we can experiment with univalence and which has a universe which is an object of itself (i.e. such that -| U:U is a valid sequent) is an experiment with combining inconsistency with univalence.

This also makes the system to be much easier to implement.

In addition, building various proofs of inconsistency in such a system provides a good testing ground for implementations.

Of course adding a U:U universe is only one way in which inconsistency helps to make things more simple.

Another way is to have inductive types and match/fixpoint machinery with weaker than what is necessary for consistency termination rules.

This relates to the currently ongoing controversy about termination checking in Coq and Agda.

Maybe we can have different “modes” there? A mode with fast computation which is known to be inconsistent (“Type in Type” and lax termination checking) and a mode with slow computation which is known to be consistent (sequence of universes and eliminators)?

And then we can study the examples of proofs which are too slow in the slow mode and based on these examples find ways to extend or modify the slow mode while preserving its consistency?

This would allow us to use some of the advantages of the LCF-style approach.

For example, HOL Light is a system with two modes like this - the fast mode is OCaml, the slow mode is the equality version of Church’s “simple theory of types”.

Another context where the question of allowing the possibility of non-terminating computation arises is the context of adding substitutional (“extensional”, “strict”) identity types to polymorphic (having a universe) systems.

A system of identity types is a construction which for every two objects “x1 x2” of a type “T” produces a type “Id T x1 x2”.

Note that this is the second, after the universe, way of producing types which depend on objects.

Let’s say that “Id” is a substitutional system of identity types if for derivable sequents

“Gamma |- e : Id U T1 T2”
“Gamma |- t : T1”

the sequent

“Gamma |- t : T2” is derivable.

In a system with substitutional identity types one can build a non-normalizable well-typed lambda-term:

T:U, e:Id U T (T->T) |- (lambda x:T, x x) (lambda x:T, x x) : T .

Strict Prop (a type of objects whose structure is never used in computation).