Some contributions of Vladimir Voevodsky to Dependent Type Theory

Thierry Coquand

Nijmegen, 22 January 2018

Plan of the talk

The TYPES community has had a quite unique opportunity to get inputs from one the top world-class mathematician

It is quite appropriate to try to analyse what has been his influence

What I will present is a first attempt, (necessarily) very incomplete

Plan of the talk

I will try to describe how was dependent type theory before Voevodsky's contributions

I will then present 3 contributions

(1) formulation of homotopy levels, equivalence and univalence axiom

(2) UniMath

(3) model in simplicial sets

In the last part, I will try to formulate some research questions motivated by these contributions

Dependent type theory

Any formal system for mathematics should address the two questions of representation of

-collection of mathematical objects: algebraic, ordered mathematical structures (Bourbaki), then collections of such structures (categories), then collections of such collections (2-categories), and so on . . .

-laws of identification of mathematical objects

How is this done in dependent type theory?

Universes and equality types (the two sources of type dependencies)

Main problem: *what should the equality mean for the universe*

Collections

We represent collection of mathematical structures using the notion of *universes*, combined with the notion of dependent sum types

E.g. $\Sigma(X : U)\ X \times (X \to X)$ type of elements (X, a, f) small type X with a constant a and a function $f : X \to X$

Collections

This notion was already present in Automath 1968, with the sort **type**

This notion is *not* present in simple type theory (HOL)

This notion is present in set theory, in the form of Grothendieck universes

A commulative hierarchy of universes was introduced by P. Martin-Löf (1973), as well as the analogy with Grothendieck universes

Collections

If we want to represent algebraic structures we need a notion of *equality*

Same problem if we want to represent the notion of *category*, we need at least a notion of equality of morphisms

In dependent type theory, there is already a notion of equality: *definitional* equality (which contains at least β, η-equality)

But this notion cannot be formulated as a type

So we need a different notion of equality, expressed as a type

Equality

This distinction between these two notions of equality was already present in the system Automath: *book* equality versus *definitional* equality

One of the first example in the first published paper (written 1968) on Automath presents exactly equality as a type!

However, there, it is presented with *axioms*

Equality

One key aspect of Martin-Löf type theory was the addition of the notion of *inductive types*, with associated computation rules (which "justifies" the introduction of this type)

This addition of data types and computation rules is an important addition, since it allows us to think of dependent type theory as a *programming* language

This programming language can be seen as an appropriate candidate for *total* functional programming (D. Turner): we have decidable type checking and canonicity

Equality as an inductive type

Inductive types: already in the version with a type of all types "On the strength of intuitionistic reasoning" (talk at Bucharest, 1971)

The language of the theory is richer than the language of first order predicate logic. This makes it possible to strengthen the axioms for existence and disjunction. In the case of existence, the possibility of strengthening the usual elimination rule seems first to have been indicated by Howard 1969. (1972, published 1995)

This makes it possible to strengthen the axioms for existence, disjunction, absurdity and identity. (1973)

Equality as an inductive type

The unique constructor is $\mathsf{ref}\ x$ of type $\mathsf{Id}\ A\ x\ x$ for $x : A$

The "uaual" elimination rule would be (least reflexive relation)

$\Pi(x : A)(y : A)\ \mathsf{Id}\ A\ x\ y \to C(x, y)$ given $d : \Pi(x : A)C(x, x)$

The *dependent* elimination rule is

$J(d) : \Pi(x : A)(y : A)(p : \mathsf{Id}\ A\ x\ y)C(x, y, p)$ given $d : \Pi(x : A)C(x, x, \mathsf{ref}\ x)$

computation rule $J(d)\ x\ x\ (\mathsf{ref}\ x) = d\ x : C(x, x, \mathsf{ref}\ x)$

Problems with this approach

Not so “canonical”: variation (Christine Paulin-Mohring)

$J'(d) : \Pi(y:A)(p:\mathsf{Id}\ A\ x\ y)\ C(y,p)$ given $d : \Pi(x:A)C(x,\mathsf{ref}\ x)$

computation rule $J'(d)\ x\ (\mathsf{ref}\ x) = d\ x : C(x,\mathsf{ref}\ x)$

Which one should one choose J or J'?

Problems with this approach

Without identity types, all rules of type theory are *invariant under observational behaviour*, which is essential for *modularity* (Th. Altenkirch)

It should be the case that if If f and g in $N \rightarrow N$ are obervationally equal and $C(f)$ is provable then also $C(g)$

This is *not* the case when we extend type theory with the inductively defined equality

E.g. if $C(x)$ is Id $(N \rightarrow N)\ f\ x$

This seems to be a strong argument against this notion of identity

Alternative approach with setoids?

Equality on natural numbers can be defined *recursively* (with universes)

Bishop notion of sets: a collection with an equivalence relation

Works for interpreting simple type structure, but does not "really" work with dependent types

-quite complex: write $W\ A\ B$ in the setoid model (E. Palmgren)

-not clear if there is a "setoid" interpretation of universes

Alternative approach with setoids?

As an illustration, G. Gonthier in his development of the SSReflect library did not try to work with setoids

Instead, however, he relies on Hedberg's Theorem (any type with a decidable equality satisfies UIP)

The experiment was positive (and he started formalizing Feit-Thompson) but it works only for "concrete" structures, with decidable equality (which is fine for graph and finite group theory)

Observational type theory

D. Turner "A Formal Theory of Types" 1987 and talk at Båstad 1989 (with some hint that equality of types should be isomorphisms)

During the discussion, P. Martin-Löf mentionned R. Gandy's interpretation of *extensional* simple type theory in *intensional* simple type theory

Martin Hofmann *Extensional concepts in intensional type theory*, PhD thesis, 1995

Observational type theory

Setoid approach does not seem to work with universes

Th. Altenkirch "Extensional Type Theory in Intensional Type Theory", LICS 1999

Th. Altenkirch, C. McBride, W. Swierstra "Observational Type Theory, Now!", 2007

Needs to extend type theory with a type of proof irrelevant propositions (for definitional equality)

Decidable conversion and type-checking

Groupoid model

F. Lamarche "A proposal about foundations" (1991)

Influenced "The groupoid model refutes uniqueness of identity proofs", M. Hofmann and Th. Streicher, LICS 1994

The model under consideration in this paper we don't consider as a intended model it rather serves the purpose of providing the desired independence of UIP

Suggests however that *intensional identity type can be used as an "internal language" for groupoids like structure*

that was one motivation of F. Lamarche's paper

Type theory and set theory

At this point, type theory seemed an interesting alternative to set theory, but only for formalizing *concrete* structures

For representing more abstract reasoning (category theory, real numbers), HOL (with function extensionality), seemed to be preferable and even more set theory (extensionality and universes)

What should the equality mean for the universe

Univalent foundations

Notes on homotopy λ-calculus, March 2006

Notes for a talk at Standford (available at V. Voevodsky gitub repository)

Very interesting to compare this document with later developments

E.g. don't use inductively defined identity type, but axiomatizes the require properties of equality $\mathsf{eq}_A(x, y)$

Key property: any two elements in a type $\Sigma(y : A)\ \mathsf{eq}_A(x, y)$ are equal

Univalent foundations

Starting point: *First of all I want to suggest a modification of the usual thesis stating that categories are higher level analogs of sets. We will take a slightly different position. We will consider groupoids to be the next level analogs of sets*

Stratification: point, propositions, sets and isomorphisms, groupoids and equivalence, . . .

Univalent foundations

The key argument for this modification of the basic thesis is the following observation - not all interesting constructions on sets are functorial with respect to maps but they are all functorial with respect to isomorphisms.

In type theory, all statements are invariant by equivalence

A very strong *modularity* property!

Type theory may be a convenient alternative to set theory, for *mathematical* reasons

Univalent foundations

The surprise is that this stratification is very simply expressed in type theory, as well as the notion of equivalence

$$\text{isContr } A \;=\; \Sigma(x : A)\Pi(y : A) \text{ Id } A\ x\ y$$

$$\text{isProp } A \;=\; \Pi(x\ y : A) \text{ isContr}(\text{Id } A\ x\ y)$$

$$\text{isSet } A \;=\; \Pi(x\ y : A) \text{ isProp}(\text{Id } A\ x\ y)$$

$$\text{isGroupoid } A \;=\; \Pi(x\ y : A) \text{ isSet}(\text{Id } A\ x\ y)$$

$$\text{isEquiv } f \;=\; \Pi(y : B) \text{ isContr}(\Sigma(x : A) \text{ Id } B\ (f\ x)\ y)$$

Really suprising that we can capture in such a simple and uniform way the notions of logical equivalence, isomorphisms, (categorical) equivalence, ...

Univalent foundations

Extending this stratification we may further consider 2*-groupoids with structures,* n*-groupoids with structures and* ∞*-groupoids with structures.* Thus a proper language for formalization of Mathematics should allow one to directly build and study groupoids of various levels and structures on them. *A major advantage of this point of view is that unlike* ∞*-categories, which can be defined in many substantially different ways the world of* ∞*-groupoids is determined by Grothendieck correspondence (see Grothendieck 1997) , which asserts that* ∞*-groupoids are "the same" as homotopy types. Combining this correspondence with the previous considerations we come to the view that not only homotopy theory but* the whole of Mathematics is the study of structures on homotopy types.

UniMath library

From this perspective a category is an example of a groupoid with structure which is rather similar to a partial ordering on a set.

Univalent foundations

The univalence axiom states that the canonical map transforming an equality to an equivalence is itself an equivalence

Elegant (and formally simple!) generalization of the axiom of *propositional extensionality* (simple type theory, Church 1940) that states that two equivalent propositions are equal

But it is expressed directly for *all* types

From the groupoid model, one would have expected a stratification U_0 for sets, U_1 for groupoids, U_2 for 2-groupoids, $\dots$

What was surprising here is that all universes $U_0,\ U_1,\ \dots$ can all have types of arbitrary h-levels

UniMath Library

Described in “An experimental library of formalized Mathematics based on the univalent foundations” (Voevodsky, Mathematical Structure in Computer Science, 2013)

Actually formalized!

Dependent types are *really* used

Frst time the dependent elimination $\Pi(x : N_0)C(x)$ is used (where N_0 is the empty type; to prove that N_0 is a proposition)!

UniMath library

New kind of mathematics

For instance, the description of the long exact sequence of homotopy groups associated to a fibration

I remember very clearly the experience of going through Vladimir's univalent foundations library files during the Christmas holidays in December 2010. The library had almost no comments, so I decided to compile it step-by-step, understanding the proofs and making notes. It was an exhilerating experience and I had the feeling of discovering gems every day: homotopy fibers, equivalences, univalence, function extensionality from univalence, h-levels, etc... (N. Gambino)

UniMath library

For a recent example, one can look at the representation of the notion of triangulated category (Tomi Pannila)

This involves reasoning about abstract abelian categories and it will be in this kind of examples that the univalent approach can be tested for a formalization of mathematics

Model in Simplicial Sets

Kan simplicial sets are used in mathematics as a possible precise definition of the notion of ∞-groupoids

Voevodsky was able to show that they form a model of dependent type theory with dependent sums, products and universes, and which furthermore satisfies the univalence axiom (using help from A.K. Bousfield)

Related works

Various homotopy model of identity types (Awodey, Warren, Garner, van den Berg, Lumsdaine, ...), but they did not deal with dependent sums and products and universes, (model of type theory with only identity types)

Warren's thesis describes a truly higher-dimension model of dependent sums, products and universes but with *strict* ∞-groupoids

Related works

Hofmann-Streicher (1995) suggested extension to weak ∞-groupoid,

However, it might be interesting to view equivalent categories as propositionally equal. This, however, would require "2-level groupoids" in which we have morphisms between morphisms and accordingly the identity sets are not necessarily discrete. We do not know whether such structures (or even infinite-level generalisations thereof) can be sensibly organised into a model of type theory.

Streicher (2016) model in simplicial sets, does not cover universes and problems with coherence for interpreting elimination rules of identity types

Model in Simplicial Sets

Even the proof that Kan simplicial sets form a model of type theory with dependent products relies on non trivial properties

The simplicial set model is *very* mysterious, since the notion of Kan fibration is a *property* rather than a *structure*, but we do need a *structure* in order to interpret the elimination rule for identity!

This issue is solved by selecting a particular structure for the universe and then to have a structure on small fibrations by pull-back

This is *not* compositional (e.g. the structure for a dependent product is not a function of the structure on the components)

Problem 1

Inspired by the simplicial model, we have found a class of *constructive* models, "cubical" type theory where Kan operation a *structure* and not a *property*

Ch. Sattler has extended this to a Quillen model structure (notion of fibrations, cofibrations and weak equivalences between any two presheaves not necessarily fibrant ones)

Can we capture in this way the Quillen model structure on simplicial sets/ CW complexes??

Problem 2

It is possible to give a semantics of higher inductive types in a constructive setting, but this is done in a "cubical" way

Can we adapt this to provide a semantics of higher inductive types in Voevodsky's simplicial set model?

Not clear yet (even in a classical framework)!

Problem 3

Propositional resizing

Extension of simple type theory (impredicative type of propositions)

Is this consistent?

If we extend cubical type theory, do we still have canonicity?

Classically, we only have a model of a weak form of propositional resizing

Problem 4

These questions of "collections" and "identifications" occur as purely mathematical questions when one tries to represent collections of mathematical structures in *sheaf* models

In set theory, universe of small sets is still a set

The universe of all small sheaves is not a sheaf!

Motivation for the notion of stacks, and then higher topos

topos theory is too much set theoretic ... *the notion of higher topos is much nicer* (M. Kontsevich, during a general discussion on topos theory)

Problem 4

This suggests various extensions of dependent type theory

There also, the models using simplicial sets only justify a weak form of universes, where, e.g. the required equality between $El\ (\pi\ a\ b)$ and $\Pi(x : El\ a)\ El\ (b\ x)$ is only an isomorphism

(This works well in a cubical setting, however, even with universes à la Russell)

Problem 5

Cubical type theory provides computation rules for univalence but for a suitable *presheaf* extension of type theory (higher order objects)

Is it possible to extend ordinary type theory with suitable computation rules which "explain" the univalence axiom in an effective way?