Type Theory and Univalent Foundations

Thierry Coquand

University of Gothenburg

Univalent Foundations of Mathematics

New axiom in mathematics: the axiom of univalence

A generalization of the axiom of extensionality (first axiom of set theory) which states that "isomorphic structures are equal"

Univalent Foundations of Mathematics

New connections between logic and topology

Some basic laws of homotopy theory (Poincaré *Analysis Situs*, 1895) have a *purely logical* explanation

Univalent Foundations of Mathematics

New connections between logic and topology

Some basic laws of homotopy theory (Poincaré *Analysis Situs*, 1895) have a *purely logical* explanation

New foundations?

Content

First lecture: Type Theory Before Voevodsky

The theory of types with which we shall be concerned is intended to be a full scale system for formalizing intuitionistic mathematics as developed, for example, in the book of Bishop 1967 (Martin-Löf, 1975)

The axiom of univalence will be expressed in type theory

This axiom is also incompatible with set theory

First Part: History, Type Theory B.V.

Important components

(1) Propositions as Types, Proofs as Elements, *new* logical operations

(2) Inductive definitions, computations (functional programming language)

(3) Universes

(4) Identity types

First Part: History, Type Theory B.V.

Successive refinements, each bringing new notions

Howard (1969) strong existence

Martin-Löf (1970-1975) strong notion of existence (axiom of choice), of disjunction of absurdity, of identity, refinement of Gentzen's introduction/elimination rules

Martin-Löf (1970-1975) universe, first a type of a all types, then as a reflection principle to express something like Grothendieck universes

Notations

The system is based on λ-calculus

$\lambda x.t$ for lambda abstraction

$t(x = a)$ for substitution, or even $t(a)$ if x is clear

$f\ a$ for application of f to a and $f\ a_1\ a_2$ for $(f\ a_1)\ a_2$

$A_0 \to A_1 \to A_2$ for $A_0 \to (A_1 \to A_2)$

Type Theory

First version (1971)

Terms $t, u, A, B \ ::= \ x \mid V \mid \lambda x.t \mid t\ u \mid (\Pi x : A)B$

$$\frac{}{\vdash} \qquad \frac{\Gamma \vdash A : V}{\Gamma, x : A \vdash} \qquad \frac{\Gamma \vdash \quad (x : A) \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma, x : A \vdash B : V}{\Gamma \vdash (\Pi x : A)B : V} \qquad \frac{\Gamma \vdash}{\Gamma \vdash V : V}$$

$$\frac{\Gamma \vdash t : (\Pi x : A)B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B(u)} \qquad \frac{\Gamma, x : A \vdash v : B}{\Gamma \vdash \lambda x.v : (\Pi x : A)B}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash B : V \quad A = B}{\Gamma \vdash t : B}$$

Type Theory

Type = Proposition

$A \to B$ defined as $(\Pi x : A)B$ where x is not free in B

Terms and types (propositions) are treated uniformely

Type Theory

The relation $\vdash a : A$ is more fundamental in intuitionism than simply asserting that A is provable

Type = Proposition Element = Proof

The relation $\vdash a : A$ should be decidable

"We recognize a proof of a proposition when we see one" (Kreisel)

Type Theory

This relation $a : A$ is *not* expressed as a type of the formal system

We cannot (directly) reproduce Russell's paradox of the set of all sets that do not contain themselves : we cannot form the proposition/type that x is not of type x

$t : A$ is a *judgement* and not a *proposition*

Types are not *Sets*

Type Theory

No term t in normal form such that

$$X : V \vdash t : X$$

This can be proved in a purely combinatorial way, and expresses some form of (weak) consistency

New connectives

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. (1972)

The language of the theory is richer than the language of traditional systems in permitting proofs to appear as parts of the propositions so that the propositions can express properties of proofs (and not only individuals, like in first-order logic). This makes it possible to strengthen the axioms for existence, disjunction, absurdity and identity. (1975)

Dependent sum

$$\frac{\Gamma, x : A \vdash B : V}{\Gamma \vdash (\Sigma x : A)B : V}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash (a, b) : (\Sigma x : A)B}$$

$$\frac{\Gamma \vdash c : (\Sigma x : A)B}{\Gamma \vdash c.1 : A} \qquad \frac{\Gamma \vdash c : (\Sigma x : A)B}{\Gamma \vdash c.2 : B(c.1)}$$

$$(a, b).1 = a \qquad (a, b).2 = b$$

Dependent sum

The new quantification operation $(\Sigma x : A)B$ was introduced by Howard (1969)

$(\Sigma x : A)B$ can be read as the proposition: there *exists* an object x of type A such that $B(x)$

Another interpretation is the type of all objects x of type A *such that* $B(x)$ *because, from the intuitionistic point of view, to give an object* x *of type* A *such that* $B(x)$ *is to give* x *together with a proof of the proposition* $B(x)$

The rule for second projection $c.2 : B(c.1)$ does not correspond to any rule in natural deduction

Disjoint union

Usual rule for disjunction

introduction rules

$$\frac{A}{A \vee B} \qquad \frac{B}{A \vee B}$$

elimination rule

$$\frac{A \to C \qquad B \to C}{A \vee B \to C}$$

Gentzen natural deduction: *introduction rules* give the meaning of logical connectives, the *elimination rules* are then justified in term of the introduction rules by suitable *normalization rules*

Disjoint union

In type theory $A + B : V$ if $A : V$ and $B : V$

Introduction rules:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{Inl}\ t : A + B} \qquad \frac{\Gamma \vdash u : B}{\Gamma \vdash \mathsf{Inr}\ u : A + B}$$

Elimination rules:

$$\frac{\Gamma \vdash a : (\Pi x : A)C(\mathsf{Inl}\ x) \qquad \Gamma \vdash b : (\Pi y : B)C(\mathsf{Inr}\ y)}{\Gamma \vdash \mathsf{Elim}\ a\ b : (\Pi z : A + B)C}$$

Computation rules:

$$\mathsf{Elim}\ a\ b\ (\mathsf{Inl}\ t) = a\ t \qquad \mathsf{Elim}\ a\ b\ (\mathsf{Inr}\ u) = b\ u$$

This is a *refinement* of Gentzen's elimination rules

Natural numbers

N is a type of *constructors* $0 : N$ and $\mathsf{S}\ x : N$ if $x : N$

Let $C(n)$ be a function which to an arbitrary natural number $n : N$ assigns a type

Given $d : C(0)$ and $e : (\Pi n : N)\ (C(n) \to C(\mathsf{S}\ n))$

we may introduce a function $f : (\Pi n : N)\ C(n)$ by the equations

$f\ 0 = d \qquad\qquad f\ (\mathsf{S}\ n) = e\ n\ (f\ n)$

Inductive definitions

We can introduce the type N_1 with only one constructor $0 : N_1$

If $C(x)$ type for $x : N_1$ and we have $a : C(0)$

we can introduce $f : (\Pi x : N_1)\ C(x)$ by the equation

$$f\ 0 = a$$

Inductive definitions

We can introduce the type N_2 with two constructors $0 : N_2$ and $1 : N_2$

If $C(x)$ type for $x : N_2$ and we have $a_0 : C(0)$ and $a_1 : C(1)$

we can introduce $f : (\Pi x : N_2)\ C(x)$ by the equations

$$f\ 0 = a_0 \qquad\qquad f\ 1 = a_1$$

Inductive definitions

We introduce the type N_0 with *no* constructor

If $C(x)$ type for $x : N_0$

we have $f : (\Pi x : N_0)\ C(x)$

$N_0 = \bot$ as a proposition

This is a *refinement* of the usual elimination rule $N_0 \to A$ for any type A

Inductive definitions

N, N_2, N_1, N_0 and $A + B$ are examples of type introduced by *ordinary inductive definition*

Hilbert 1926 considers the type *Ord* of *ordinal numbers*

$0 : Ord$
if $x : Ord$ then $\mathsf{S}\ x : Ord$
if $u : N \to Ord$ then its limit $L\ u : Ord$

One can write elimination rules for this type as well (Martin-Löf, 1971)

Inductive definitions

All these constructions follow the same pattern

constructors correspond to primitive operations and to introduction rules in natural deduction

defined functions correspond to elimination rules that are justified by

computation rules which correspond to normalization in natural deduction

Computation rules correspond to the process of understanding a proof by eliminating lemmas

Type theory

Type theory can be seen as a *functional programming language* with dependent types

"Extension" of PCF (but only *total* functionals)

Terms of type theory: λ-calculus with constants

Two kind of constants: *constructors* or *defined functions*

Type theory

A term is *normal* if it cannot be further reduced

The closed normal term of type N_2 are exactly 0 and 1

The closed normal term of type N have the form $0,\ \mathsf{S}(0),\ \mathsf{S}(\mathsf{S}(0)), \ldots$

There is no normal closed term of type N_0, since there is no canonical term of type N_0 and normal closed term are canonical

Hence *if* we have normalization there is *no* closed term of type N_0 since types are preserved by reduction

Since $\bot = N_0$ this expresses the *logical consistency* of type theory

Type theory

In the formal theory the abstract entities (natural numbers, ordinals, functions, types, and so on) become represented by certain symbol configurations, called terms, *and the definitional schema, read from the left to the right, become mechanical* reduction rules *for these symbol configurations.*
Type theory *effectuates the computerization of abstract intuitionistic mathematics that above all Bishop has asked for* (1971)

Type Theory provides a framework in which we can express *conceptual* mathematics in a *computational* way.

Type theory

The law of excluded middle EM becomes

$$(\Pi X : V)(X + \neg\, X)$$

where $\neg\, X$ is $X \to N_0$

Another equivalent formulation is

$$(\Pi X : V)(\neg(\neg X) \quad \to X)$$

There is no term in normal form of this type

We represent *intuitionistic logic* and type theory can be seen as a concrete representation of Kolmogorov 1930 interpretation of this logic

The W-type

Type of trees T (P. Aczel)

A tree $t : T$ should be given by an index type $I : V$ with a family of trees $f : I \to T$

Generalization: given any type family $B(x)$ $(x : A)$ we define $(\mathrm{W}x : A)B$ by constructor

$$\sup\ a\ f : (\mathrm{W}x : A)B$$

if $a : A$ and $f : B(a) \to (\mathrm{W}x : A)B$

We can deduce the induction principle

The W-type

New logical operation

$(\Pi x : N_2)B$ is equivalent to $B(0) \wedge B(1)$

$(\Sigma x : N_2)B$ is equivalent to $B(0) + B(1)$

$(\mathrm{W}x : N_2)B$ is equivalent to $\neg(B(0) \wedge B(1))$

Sets as Trees

We can represent *sets* as elements of the type $S = (\mathrm{W}X : V)X$

A set $t = \sup\ I\ f : S$ is given by an index type $I : V$ and a family of sets $f : I \to V$

Equality ($=$ bissimulation) and membership are functions of types $S \to S \to V$ defined by induction

V is used both as type of types and type of propositions

The type of types

The idea of the type of types is forced upon us by accepting simultaneously each of the following three principles.

First, quantification over propositions as in second order logic.

Second, Russell's doctrine of types according to which the ranges of significance of propositional functions form types so that, in particular, it is only meaningful to quantify over all objects of a certain type.

Third, the identification of propositions and types.

Suppose namely that quantification over propositions is meaningful. Then by the doctrine of types, the propositions must form a type. But, if propositions and types are identified, then this type is at the same time the type of types.

The type of types

The type of types introduces a strong kind of selfreference which, as pointed out by Gödel 1964, transcends the cumulative hierarchy notion of set and may seem to verge on the paradoxes, but which is actually being used in category theory, notably, in the construction of the category of all categories

The type of types

We can define types by recursion, for instance $T : N \to V$

$T\ 0 = N \qquad T\ (n+1) = (T\ n) \to N$

but also $Eq : N \to N \to V$

$Eq\ 0\ 0 = N_1 \qquad Eq\ (n+1)\ (m+1) = Eq\ n\ m$

$Eq\ 0\ (m+1) = Eq\ (n+1)\ 0 = N_0$

Girard's paradox

It does not seem possible to translate directly Russell's paradox

We can form $(\Sigma X : V)B(X)$ but $B(X)$ cannot express that X is not of type X

The judgement $a : A$ is not a proposition that can be negated

Girard's paradox

Using the type $S = (\mathrm{W}X : V)X$ we can interpret Russell's paradox since we have $S : V$ and the tree $u = \sup\ S\ (\lambda x.x)$ represents a set of all sets

Actually in this representation the membership relation has to be well-founded and we have also a representation of Burali-Forti's paradox

We have a closed term t such that

$$\vdash t : \quad \bot \quad = \quad N_0$$

and this closed term t is *not* normalizable

Universe

The incoherence of the idea of a type of all types whatsoever made it necessary to distinguish, like in category theory, between small and large types. Thus the universe U appears, not as the type of all types, but as the type of small types, whereas U itself and all types which are built up from it are large. This makes the types wellfounded and the theory predicative.

Universe

This is reminiscent of Grothendieck's notion of universe

If $A : U$ and $B : U$ for $x : A$ then we have

$$(\Pi x : A)\ B : U \qquad\qquad (\Sigma x : A) B : U$$

$N_0,\ N_1,\ N_2, N$ are all of types U
We can define small types by recursion

Type Theory with one universe (1972)

Terms $t, u, A, B \ ::= \ x \mid U \mid \lambda x.t \mid t\ u \mid (\Pi x : A)B$

$$\frac{}{\vdash} \qquad \frac{\Gamma \vdash A}{\Gamma, x : A \vdash} \qquad \frac{\Gamma \vdash \quad (x : A) \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\Gamma \vdash}{\Gamma \vdash U} \qquad \frac{\Gamma, x : A \vdash B}{\Gamma \vdash (\Pi x : A)B}$$

$$\frac{\Gamma \vdash t : (\Pi x : A)B \quad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B(u)} \qquad \frac{\Gamma, x : A \vdash v : B}{\Gamma \vdash \lambda x.v : (\Pi x : A)B}$$

$$\frac{\Gamma \vdash t : A \quad \Gamma \vdash B \quad A = B}{\Gamma \vdash t : B}$$

$$\frac{\Gamma \vdash A : U \quad \Gamma, x : A \vdash B : U}{\Gamma \vdash (\Pi x : A)B : U} \qquad \frac{\Gamma \vdash A : U}{\Gamma \vdash A}$$

Type Theory with one universe (1972)

We can define $F : N \to U$

$$F\ 0 = N \qquad F\ (n+1) = (F\ n) \to N$$

In ZF set theory, this function can only be defined with the *axiom schema of replacement*

Consistency and normalization

There is no proof in normal form of $\bot = N_0$, since a closed normal term is in canonical form

Captures elegantly the essence of consistency arguments in proof theory

The notion of universe (1972) was extended with a cumulative hierarchy of universes $U_0 : U_1 : U_2 : \ldots$

Equality types

The language of the theory is richer than the language of traditional systems in permitting proofs to appear as parts of the propositions so that the propositions can express properties of proofs (and not only individuals, like in first-order logic). This makes it possible to strengthen the axioms for existence, disjunction, absurdity and identity. (1975)

Equality in Logic and Mathematics

Frege (1879) and Russell (1903) define $x = y$ as

$$\forall P.\ P(x) \leftrightarrow P(y)$$

Leibnitz's identity of indiscernibles

Equality in Type Theory

Introduced in 1975

Strengthening of usual laws of equality

$$\frac{}{x = x} \qquad \frac{x = y \qquad \forall z.C(z, z)}{C(x, y)}$$

$$\frac{}{x = x} \qquad \frac{x = y \qquad P(x)}{P(y)}$$

Equality in Type Theory

In type theory

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{Ref}\ t : \mathsf{Id}_A\ t\ t}$$

$$\frac{\Gamma \vdash d : (\Pi x : A) C(x, x, \mathsf{Ref}\ x)}{\Gamma \vdash \mathsf{J}\ d : (\Pi x_0\ x_1 : A)(\Pi p : \mathsf{Id}_A\ x_0\ x_1) C(x_0, x_1, p)}$$

$$\mathsf{J}\ d\ x\ x\ (\mathsf{Ref}\ x) = d\ x : C(x, x, \mathsf{Ref}\ x)$$

Equality in Type Theory

Variation of the rules for equality

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathsf{Ref}\ t : \mathsf{Id}_A\ t\ t}$$

$$\frac{\Gamma \vdash d : C(a, \mathsf{Ref}\ a)}{\Gamma \vdash \mathsf{J}'\ d : (\Pi x : A)(\Pi p : \mathsf{Id}_A\ a\ x)C(x, p)}$$

$$\mathsf{J}'\ d\ a\ (\mathsf{Ref}\ a) = d : C(a, \mathsf{Ref}\ a)$$

Equality in Type Theory

New logical rules

We can iterate the formation of identity types

What is the meaning of $\mathsf{Id}_{\mathsf{Id}_A\ a\ u}\ \alpha\ \beta$??

Are such types interesting?

Equality in Type Theory

Do we have Uniqueness of Identity Proofs?

Otherwise, what is the meaning of an equality between equality proofs?

Answer(?): homotopy theory