Univalent Type Theory

Thierry Coquand

Tutorial for the Logic Colloquium 2016, Leeds

Univalent type theory

- stratified notion of ⟪being the same⟫ for mathematical collections (at least isomorphisms and categorical equivalences)

- one goal: to design a formalism in which it is impossible to formulate a statement which is not invariant with respect to equivalences

- simple type theory and connections with set theory

- limitations of simple type theory, addition of universes

- introduction of equality as a *type family*

- what should be the axiom of extensionality for this universe?

New laws for equality

Id A a_0 a_1 can be thought of as the type of ⟪identifications⟫ of a_0 and a_1

Intuitively, if A is a set there is at most one such identification

If A is the collection of all sets, an identification is a bijection

If A is a collection of structure, an identification is an isomorphism

Martin-Löf expressed Id A a_0 a_1 as a type since he wanted to develop systematically the notion of propositions as types

Similarly, de Bruijn had a notion of ⟪book equality⟫, but his motivation was to design a good way to represent mathematical proofs in a computer

New laws for equality

If $B(x)$ is a family of types over A then

any $p : \mathsf{Id}\ A\ a_0\ a_1$ defines a transport function $t(p) : B(a_0) \to B(a_1)$

For instance, if A is the collection of sets, and $B(X)$ is the collection $X \to X$ then any isomorphism $p : X_0 \simeq X_1$ defines a transport function

$B(X_0) \to B(X_1)$

$t(p)\ u_0 = p \circ u_0 \circ p^{-1}$

This was the notion of transport of structures considered by Bourbaki

New laws for equality

How can we have a formal system with Id U N Q where N type for natural numbers and Q for rational numbers?

Don't we have $2/3 \in Q$ and $0 \in N$?

This is where the use of type theory is important

The statement $2/3 \in Q$ is *not* a well-formed proposition in type theory

Types and abstraction

Story of J. Reynolds

One semester, large parallel course in complex variables with two sections

-in one section, Professor Descartes, $z = x + iy$

-in one section, Professor Bessel, $z = \rho e^{i\theta}$

《After presenting the definitions of complex numbers, both went on in explaining how to convert reals into complex numbers. how to add, multiply, and conjugate complex numbers, and how to find their magnitude》

Types and abstraction

《 Then, after their first classes, an unfortunate mistake in the register's office cause the two sections to be interchanged 》

No problem however!

《 The reason was that they both had an intuitive understanding of type. Having defined complex numbers and the primitive operations upon them, thereafter they spoke at a level of abstraction that encompossed both of their definitions 》

The moral of the story is

Type structure is a syntactic discipline for enforcing levels of abstraction

New laws for equality

We have an element in $\mathsf{Id}\ S\ (a, 1_a)\ (x, p)$

where $S = (x : A) \times \mathsf{Id}\ A\ a\ x$ and $x : A$ and $p : \mathsf{Id}\ A\ a\ x$

Does not this imply that we have equality of 1_a and p?

Equality in 《sigma》 types is subtle!

New laws for equality

In set theory if we have $(a_0, b_0) = (a_1, b_1)$ (in *any* set) we get

$a_0 = a_1$ and $b_0 = b_1$

In a type $(x : A) \times B(x)$ what does the equality of (a_0, b_0) and (a_1, b_1) mean?

We can form $\mathsf{Id}\ A\ a_0\ a_1$ since a_0 and a_1 are of type A

But b_0 is of type $B(a_0)$ and b_1 is of type $B(a_1)$, so we cannot compare them!

If $p : \mathsf{Id}\ A\ a_0\ a_1$ we have a transport function $t(p) : B(a_0) \to B(a_1)$

We ask for a proof of equality of $t(p)\ b_0$ and b_1, both are in $B(a_1)$

New laws for equality

So an equality between (a_0, b_0) and (a_1, b_1) is intuitively given by an equality p : Id $A\ a_0\ a_1$ and an equality ptoof in Id $B(a_1)\ (t(p)\ b_0)\ b_1$

New laws for equality

Semantically let us look at the example S the type of sets

$T(X)$ is the set X^A

We want to understand the «equality» in $\sum_{X:S} T(X)$

If $u : X_0 \simeq X_1$ we have a transport function

$t(u) : T(X_0) \to T(X_1)$

$t(u)\ f_0 = u \circ f_0$

And an equality $(X_0, f_0) \simeq (X_1, f_1)$ will be given by $u : X_0 \simeq X_1$ such that $u \circ f_0 = f_1$

New laws for equality

In this way, the new law discovered by Martin-Löf (1973) that we have an element in $\mathsf{Id}\ S\ (a, 1_a)\ (x, p)$

where $S = (x : A) \times \mathsf{Id}\ A\ a\ x$ and $x : A$ and $p : \mathsf{Id}\ A\ a\ x$

can be understood as: we look at the transport function $t(p)$ for the family $C(z) = \mathsf{Id}\ A\ a\ z$ over A and $t(p)\ 1_a$ is equal to p

New laws for equality

Usual formulation is

$$(x : A) \to (p : \mathsf{Id}\ A\ a\ x) \to C(a, 1_a) \to C(x, p)$$

which generalizes the usual ⟪elimination⟫ rule

$$(x : A) \to \mathsf{Id}\ A\ a\ x \to P(a) \to P(x)$$

New laws for equality

Let us《explain》this law on the following example

Let S be the collection of《all》sets, seen as a groupoid

We fix a set A and define Q to be the collection $\sum_{X:S} A \simeq X$

Any element (X, f) of Q can be identified to (A, id) since $f = f \circ id$, and actually, this identification is uniquely determined

Q seen as a groupoid is equivalent to the groupoid with one object and one morphism

New laws for equality

Let us define

$\text{isContr}\ T = \qquad (t : T) \times ((x : T) \to \text{Id}\ T\ t\ x)$

This describes when a collection is ⟪equivalent⟫ to a singleton

The new law of equality can be expressed as inhabitant of

$\text{isContr}\ ((x : A) \times \text{Id}\ A\ a\ x)$

for any type A and a element of A

New laws for equality

To summarize we extend type theory with the constants

- $\mathsf{Id}\ A\ a_0\ a_1$

- $1_a : \mathsf{Id}\ A\ a\ a$

- $t(p) : B(a_0) \to B(a_1)$ if $p : \mathsf{Id}\ A\ a_0\ a_1$

- a proof of $\mathsf{Id}\ B(a)\ (t(1_a)\ u)\ u$ if $u : B(a)$

- a proof of $\mathsf{Id}\ S\ (a, 1_a)\ (x, p)$ for $S = (x : A) \times \mathsf{Id}\ A\ a\ x$ and $(x, p) : S$

New laws for equality

These laws were discovered in 1973

Should equality be extensional?

Actually, how to express the extensionality axioms in this context?

An answer to this question is given by Voevodsky (2010)

Equivalence

A simple and uniform notion of equivalence for $f : A \to B$

If A and B are *sets* we get back the notion of *bijection* between sets

If A and B are *propositions* we get back the notion of *logical equivalence* between propositions

If A and B are *groupoids* we get back the notion of *categorical equivalence* between groupoids

Equivalence

$$\mathsf{Fib}\ f\ b = \qquad (a : A) \times \mathsf{Id}\ B\ b\ (f\ a)$$

$$\mathsf{isEquiv}\ f = \qquad (b : B) \to \mathsf{isContr}\ (\mathsf{Fib}\ f\ b)$$

$$\mathsf{Equiv}\ A\ B = \qquad (f : A \to B) \times \mathsf{isEquiv}\ f$$

We recall

$$\mathsf{isContr}\ T = \qquad (t : T) \times ((x : T) \to \mathsf{Id}\ T\ t\ x)$$

Equivalence

If A is a type, let us unfold isEquiv id

$(b : A) \to \text{isContr}\ ((a : A) \times \text{Id}\ A\ b\ a)$

This is *exactly* the new law of equality discovered by Martin-Löf

So the identity function is always an equivalence

Hence we have a proof of Equiv A A

Equivalence

It follows directly from the definition of

$$\mathsf{isEquiv}\ f = (b : B) \to \mathsf{isContr}\ ((a : A) \times \mathsf{Id}\ B\ (f\ a)\ b)$$

that we have

$$\mathsf{isEquiv}\ f \quad \to \quad (B \to A)$$

Equivalence

In particular, if we define $A \leftrightarrow B$ by $(A \to B) \times (B \to A)$

$\mathsf{Equiv}\ A\ B \ \ \to \quad A \leftrightarrow B$

equivalence implies logical *equivalence*

The Univalence Axiom

The *univalence axiom* states roughly that if

$f : A \to B$

is an equivalence then A and B are equal

More exactly, since $\mathsf{Equiv}\ A\ A$ we have a map $\mathsf{Id}\ U\ A\ B \to \mathsf{Equiv}\ A\ B$

The canonical map $\mathsf{Id}\ U\ A\ B \to \mathsf{Equiv}\ A\ B$ *is an equivalence*

This generalizes Church's axiom of extensionality for *propositions*

Voevodsky has shown that this axiom implies *function extensionality*

The Univalence Axiom

If $p : \mathsf{Id}\ U\ \ A\ \ B$ we have, by defining $C(X) = \mathsf{Equiv}\ \ A\ \ X$

$t(p) : C(A) \to C(B)$

But we have a proof $q : \mathsf{Equiv}\ \ A\ \ A = C(A)$

So we have $t(p)\ \ q : C(B) = \mathsf{Equiv}\ \ A\ \ B$

This defines a function $f : \mathsf{Id}\ U\ \ A\ \ B \to \mathsf{Equiv}\ \ A\ \ B$

And the univalence axiom is that this function is an equivalence

The *statement* itself of the univalence axiom uses the representation of propositions as types

New laws for equality

- $\mathsf{Id}\ A\ a_0\ a_1$
- $1_a : \mathsf{Id}\ A\ a\ a$
- $t(p) : B(a_0) \to B(a_1)$ if $p : \mathsf{Id}\ A\ a_0\ a_1$
- a proof of $\mathsf{Id}\ B(a)\ (t(1_a)\ u)\ u$ if $u : B(a)$
- a proof of $\mathsf{Id}\ S\ (a, 1_a)\ (x, p)$ for $S = (x : A) \times \mathsf{Id}\ A\ a\ x$ and $(x, p) : S$
- the univalence axiom

New laws for equality

Voevodsky was looking for a formalism in which it is ⟪impossible to formulate a statement which is not invariant with respect to equivalences⟫

The formalism of type theory (as designed by de Bruijn, Martin-Löf, ...) is such a formalism, provided we add the univalence axiom

The Univalence Axiom

The canonical map Id $U\ A\ B \rightarrow$ Equiv $A\ B$ *is an equivalence*

We have seen that equivalence implies logical equivalence

So the univalence axiom implies

Equiv $A\ B \rightarrow$ Id $U\ A\ B$

but it is much more subtle

Equivalence and 《isomorphism》

If we define hasInv f to be the type

$(g : B \to A) \times \ \mathsf{Id}\ (B \to B)\ (f \circ g)\ id\ \times\ \mathsf{Id}\ (A \to A)\ (g \circ f)\ id$

we have

isEquiv $f \to$ hasInv f

⟪Grad Students⟫ Lemma

Lemma: hasInv $f \to$ isEquiv f

If we define

Iso A $B = (f : A \to B) \times$ hasInv f

we get

(Equiv A B) $\leftrightarrow$ Iso A B

The Univalence Axiom

$\mathsf{Id}\ U\ (A \times B)\ (B \times A)$

$\mathsf{Id}\ U\ (A \times (B \times C))\ ((A \times B) \times C)$

Any property satisfied by $A \times B$ that can be expressed in type theory is also satisfied by $B \times A$

This is not the case in set theory

$(1, -1) \in \mathbb{N} \times \mathbb{Z} \qquad\qquad (1, -1) \notin \mathbb{Z} \times \mathbb{N}$

Stratification of types

A type A is a *proposition*

$(x_0\ x_1 : A) \to \mathsf{Id}\ A\ x_0\ x_1$

Notice that this itself is a type

A type is a *set*

$(x_0\ x_1 : A) \to \mathsf{isProp}(\mathsf{Id}\ A\ x_0\ x_1)$

A type is a *groupoid*

$(x_0\ x_1 : A) \to \mathsf{isSet}(\mathsf{Id}\ A\ x_0\ x_1)$

Stratification of types

The notions of *propositions*, *sets*, *groupoids* have now aquired a precise meaning in type theory

They will be used with this meaning in the rest of this tutorial

Type theory appears as a generalization *of set theory*

This stratification corresponds to the informal stratification of collection of mathematical objects that was described at the beginning of the talk

The Univalence Axiom

This axiom also implies that

-two isomorphic sets are equal

-two isomorphic algebraic structures are equal

-two equivalent (in the categorical sense) groupoid are equal

-two equivalent categoruies are equal

The equality of a and b entails that any property of a is also a property of b

The Univalence Axiom

If A and B are *propositions*, we shall see that $A \to A$ and $B \to B$ are also propositions, so we have proofs of

$\mathsf{Id}\ (A \to A)\ (g \circ f)\ id \qquad \mathsf{Id}\ (B \to B)\ (f \circ g)\ id$

for any $f : A \to B$ and $g : B \to A$

So we have $\mathsf{hasInv}\ f$ and by the Grad Students Lemma $\mathsf{isEquiv}\ f$

By univalence $\mathsf{Id}\ U\ A\ B$

Actually we have $\mathsf{Equiv}\ (\mathsf{Id}\ U\ A\ B)\ (A \leftrightarrow B)$

Univalence axiom implies Church's extensionality axiom for propositions

Motivation for the term 《proposition》

N.G. de Bruijn introduced the notion of *proof irrelevance*

His example was the following

If we want to represent the logarithm function it should be a function $log\ x\ p$ of 2 arguments

$x : R$ and p a proof that we have $x > 0$

We do not want $log\ x\ p$ to depend on p

For this, it is enough to have Id $(x > 0)\ p\ q$ for p and q are of type $x > 0$

This 《proof irrelevance》 is here used as a *definition* of the notion of proposition

Motivation for the term ⟪set⟫

If A represents a set we want $\mathsf{Id}\ A\ a_0\ a_1$ to be a proposition

At most one identification between a_0 and a_1

If A and B are sets, we can show

$\mathsf{Equiv}\ (\mathsf{Equiv}\ A\ B)\ (\mathsf{Iso}\ A\ B)$

where

$\mathsf{Iso}\ A\ B = \quad (f : A \to B) \times \mathsf{hasInv}\ A\ B\ f$

Motivation for the term «groupoid»

If A is any type we have operations of types

$1_a : \mathsf{Id}\ A\ a\ a$

$\mathsf{sym} : \mathsf{Id}\ A\ a_0\ a_1 \rightarrow \mathsf{Id}\ A\ a_1\ a_0$

$\mathsf{comp} : \mathsf{Id}\ A\ a_0\ a_1 \rightarrow \mathsf{Id}\ A\ a_1\ a_2 \rightarrow \mathsf{Id}\ A\ a_0\ a_2$

and we have e.g. for $p : \mathsf{Id}\ A\ a_0\ a_1$

$\mathsf{Id}\ (\mathsf{Id}\ A\ a_0\ a_1)\ (\mathsf{comp}\ 1_{a_0}\ p)\ p$

This uses in a crucial way the new law for equality discovered by Martin-Löf

Motivation for the term 《groupoid》

If each Id A a_0 a_1 is a set, we can think of A as a groupoid in the 《usual》 sense

An object is an element of type A

A morphism between a_0 and a_1 is an element of the *set* Id A a_0 a_1

Any morphism is an isomorphism

Difference between *properties* and *structure*

A *property* is a dependent family which is always a *proposition*

E.g. isContr A, isProp A, isSet A are properties of A

isEquiv f is a property of f

On the other hand hasInv A B f is a structure for f in general

The *univalence axiom* is stated as a *proposition*

Difference between *properties* and *structure*

For instance we can build a term of type

$(A : U) \rightarrow \mathsf{isProp}\ (\mathsf{isContr}\ A)$

All these facts correspond to knwon observations in the theory of homotopy

The fact that we can do this only using the finite list of rules about equality, mainly the new Marin-Löf law of equality and the univalence axiom, and the rules of type theory, is quite remarkable