Applications of Type Theory to Univalent Mathematics

Thierry Coquand

Autumn School, Proof and Computations, September 2019

What we saw last time

Proposition or subsingleton

$$\mathsf{isProp}\ A \quad = \quad \Pi(x:A)\Pi(y:A)\ x =_A y$$

Singleton or contractible

$$\mathsf{isContr}\ A \quad = \quad \Sigma(x:A)\Pi(y:A)\ x =_A y$$

Equivalence

$$\mathsf{isEquiv}\ f \quad = \quad \Pi(y:B)\mathsf{isContr}\ ((\Sigma x:A)\ y =_B f(x))$$

What we saw last time

$\mathsf{Refl} : \Pi(x : X)\ x =_X x$

Given $(d : \Pi(x : X) C(x, x, \mathsf{Refl}\ x))$ we have

$\mathsf{J}\ d\ :\ \Pi(x_0\ x_1 : X)(p : x_0 = x_1) C(x_0, x_1, p)$

with the "computation rule"

$\mathsf{J}\ d\ x\ x\ (\mathsf{Refl}\ x) \equiv d\ x$

What we saw last time

$\mathsf{inl} : A \to A + B$ and $\mathsf{inr} : B \to A + B$

Given $d_A : \Pi(x : A)C(\mathsf{inl}\ x)$ and $d_B : \Pi(y : B)C(\mathsf{inr}\ y)$ we have

$\mathsf{case}\ d_A\ d_B\ \ :\ \ \Pi(z : A + B)C(z)$

with the "computation rules"

$\mathsf{case}\ d_A\ d_B\ (\mathsf{inl}\ x) \equiv d_A\ x$ and $\mathsf{case}\ d_A\ d_B\ (\mathsf{inr}\ y) \equiv d_B\ y$

What we saw last time

$0 : \mathsf{Nat}$ and $\mathsf{succ} : \mathsf{Nat} \to \mathsf{Nat}$

Primitive recursion type $A \to (\mathsf{Nat} \to A \to A) \to \mathsf{Nat} \to A$ becomes

Given $a : P\ 0$ and $b : \Pi(n : \mathsf{Nat})(C(n) \to C(\mathsf{succ}\ n))$ we have

$\mathsf{natrec}\ a\ b\ \ :\ \ \Pi(n : \mathsf{Nat})C(n)$

with the "computation rules"

$\mathsf{natrec}\ a\ b\ 0 \equiv a$ and $\mathsf{natrec}\ a\ b\ (\mathsf{succ}\ n) \equiv b\ n\ (\mathsf{natrec}\ a\ b\ n)$

An important consequence

$$\Pi(x : X)\mathsf{isContr}\ (\Sigma(y : X)\ x = y)$$

We *do not have* in general $\Pi(x : A)(y : A)\mathsf{isSubsingleton}\ (x = y)$

An important consequence

Indeed, to apply Leray's theory I needed to construct fibre spaces which did not exist if one used the standard definition. Namely, for every space X, I needed a fibre space E with base X and with trivial homotopy (for instance contractible). But how to get such a space? One night in 1950, on the train bringing me back from our summer vacation, I saw it in a flash: just take for E the space of paths on X (with fixed origin a), the projection $E \to X$ being the evaluation map: path $\to$ extremity of the path. The fibre is then the loop space of (X, a). I had no doubt: this was it! $\ldots$

It is strange that such a simple construction had so many consequences.

J.-P. Serre, describing the "loop space method" introduced in his thesis (1951)

Motivations

Consider a collection of mathematical structures with isomorphisms as equality

This forms a groupoid

Fix one structure A_0

Consider the collection of structures A, f where $f : A_0 \cong A$

This forms a new groupoid which is “contractible”

Exactly one arrow between two objects

Unique up to unique isomorphism

Motivations

This example also motivates univalence in the form

$\Pi(A_0 : U)$ isContr $(\Sigma(A : U)\ A_0 \cong A)$

Motivations

Let $F : \mathcal{C} \to \mathcal{D}$ be a functor fully faithful and essentially surjective

Fix an object Y in D

Then the groupoid of pairs X, f where $f : F(X) \cong Y$ is contractible

In set theory, we need choice to be an inverse $G : \mathcal{D} \to \mathcal{C}$

This is conceptually not satisfactory

We should use some form of "unique" choice

This is captured by the notion of contractibility: strong form of uniqueness

Notion of Equivalence

In the setting of quasicategories this notion is highly non trivial

We need to define a Quillen model structure on simplicial sets

This was done by André Joyal

This model structure comes with a notion of weak equivalence

The (formally) simple description in the language of type theory gives directly some important laws that this notion should satisfy

What we saw last time

Univalence axiom

$\Pi(A_0 : U)\mathsf{isContr}\ (A_0 \cong A)$

An important consequence of univalence

Pointwise equal functions are equal

Given $A : U, B : A \to U,\ f : \Pi(x : A)B,\ g : \Pi(x : A)B$

We have

$$(\Pi(x : A)\ f\ x = g\ x) \to\ f = g$$

Extensionality Principle

In Alonzo Church simple type theory (HOL), two formulations

-function extensionality: pointwise equal functions are equal

-propositional extensionality: equivalent propositions are equal

Univalence can also be stated as the fact that the canonical map

$$A =_U B \quad \rightarrow \quad A \cong B$$

is an equivalence

This expresses "type" extensionality

It is formally remarkable that this implies function extensionality

Another example of when Curry–Howard goes wrong: $\boxed{\text{image}}$

Define the image of a function $f : X \to Y$ in the usual way, translated to Curry-Howard:

$$\text{image}\, f \equiv \Sigma(y : Y), \Sigma(x : X), f(x) = y.$$

- This is the type of points $y : Y$ for which we have some $x : X$ with $f(x) = y$.
- Trouble: $\text{image}\, f \simeq X$.

  $\boxed{\text{This is not what we expect.}}$
- Example. We don't expect the image of the unique function $\mathbb{2} \to \mathbb{1}$ to be isomorphic to $\mathbb{2}$.

  We expect the image to be a subtype of $\mathbb{1}$.

Univalent logic fixes such things in the same way as topos logic.

Image

Fiber of the projection

$(\Sigma(y : Y)\Sigma(x : X)\ y = f(x)) \quad \to X$

at x_0 in X; we obtain

$\Sigma(y : Y)\Sigma(x : X)\ (f(x) = y) \times x_0 = x$

which is equivalent to

$\Sigma(y : Y)(\Sigma z : \Sigma(x : X)x_0 = x)\ f(\pi_1\ z) = y$

equivalent to $\Sigma(y : Y)\ f(x_0) = y$ and is is contractible

Image

In the special case $X = 2$ and $Y = 1$ we obtain that

$\Sigma(y : Y)\Sigma(x : X)\ y = f(x)$

is equivalent to 2: it cannot be a possible way to represent the image

Propositional truncation (or reflection)

1. A propositional truncation of a type X, if it exists, is the universal solution to the problem of mapping X to a proposition:

$$\begin{tikzcd}
\text{arbitrary type} & X \arrow[r] \arrow[dr] & \|X\| \arrow[d, dotted] & \text{proposition} \\
& & P & \text{proposition.}
\end{tikzcd}$$

   $\|X\|$ is required to be the smallest proposition X maps into.

2. Several kinds of types can be shown to have truncations in MLTT.
3. There are a number of ways to extend MLTT to get truncations for *all* types.
   (Such as resizing $+$ funext, or higher inductive types.)

Example concluded: univalent image

The image of a function $f : X \to Y$ is

$$\mathrm{image}\, f \equiv \Sigma(y : Y), \|\Sigma(x : X), f(x) = y\|.$$

Correct formulation of unique existence

- Not $(\Sigma(x : X), A(x)) \times (\Pi(x, y : X), A(x) \times A(y) \to x = y)$.
- Instead $\mathrm{isSingleton}(\Sigma(x : X), A(x))$.

  Especially when formulating universal properties.
- A unique $x : X$ such that $A(x)$ is not enough.
- What is really needed is a unique *pair* (x, a) with $x : X$ and $a : A(x)$.

  Like in category theory again.

  Unless all types are sets.

Univalent mathematics

Σ is used to express given structure or data (cf. the type of groups)

Truncated Σ is used to express existence

Even better, one can use Σ to form propositions!

What is crucial is to have a language where we can talk in an uniform way about structure/data and propositions

The mathematical language of type theory allows this

Univalent mathematics

For instance when defining for $f : A \to B$

$$\mathsf{Iso}\ f \quad = \quad \Sigma(g : B \to A)\ (gf = 1_A) \times (fg = 1_B)$$

and

$$\mathsf{isEquiv}\ f \quad = \quad \Pi(b : B)\ \mathsf{isContr}\ (\Sigma(a : A)\ b = f(a))$$

isEquiv f is a proposition/subsingleton

Iso f is not a subsingleton in general

isEquiv f is the proposition truncation of $Iso\ f$

Univalent mathematics

$f : A \to B$ is an embedding

The canonical map $a = b \to f\ a = f\ b$ is an equivalence

Then $\Sigma(x : A) f(a) = f(x)$ is a proposition

This is equivalent to $\Pi(y : B)$ isProp $(\Sigma(x : A)\ f(x) =_B y)$

$f : A \to B$ is surjective

$\Pi(y : B)\ \|\Sigma(x : A) y = f(x)\|$

This is the essence of the result that a functor which is fully faithful and essentially surjective is an equivalence

Identification of mathematical structures

For all mathematical purposes

(1) two groups are regarded to be the same if they are isomorphic

e.g. one says that the additive integers for “the” free group on one generator

(2) two metric spaces are regarded to be the same if they are isometric

(3) two topological spaces are regarded to be the same if they are homeomorphic

(4) two categories are regarded to be the same if they are equivalent

Identification of mathematical structures

Do we *choose* the above identifications motivated by particular applications?

Or are these notions of “sameness” imposed upon us, independently of what we want to do with the structures