Univalent Foundation and Constructive Mathematics

Thierry Coquand

Luminy, April 7, 2014

References on univalent foundation

V. Voevodsky Univalent foundation home page and
"Experimental library of univalent foundation of mathematics"

B. Ahrens. C. Kapulkin, M. Shulman
"Univalent categories and the Rezk completion"

D. Grayson Foundations.Ktheory

The Univalent Foundation Program
Homotopy Type Theory: Univalent foundation of mathematics

References on constructive mathematics

R. Mines, F. Richman and W. Ruitenburg
A Course in Constructive Algebra

H. Lombardi home page

H. Lombardi and C. Quitté
Algèbre commutative: méthodes constructives

Content of the lectures

4 first lectures: introduction to univalent foundation

Roughly cover the first 4 chapters of the Univalent Foundation Program book

2 last lectures: semantics, groupoid model, presheaf models

Univalent Foundations

The goal is to design a suitable formal system for expressing and *checking* mathematical proofs on a computer

These issues have close connections with *proof theory* and *foundation of mathematics*

Traditionally, this interaction has been going in one direction

I will present a *new vision on the foundations of mathematics* suggested by the quest of a good formal system for expressing mathematical proofs

In particular, new insight on one of the most basic notion of mathematics the notion of *equality*

Current existing foundations

(1) Set theory: ZFC

$\rightarrow$ system MIZAR

(2) Category theory

Univalent Foundations

"It is extremely difficult to accept that mathematics is in need of a completely new foundation . . ."

"There is a good reason it is difficult: the existing foundation of mathematics -ZFC, and its main contender for a new foundation -category theory, have been very successful"

It is overcoming the appeal of category theory as a candidate for a new foundation of mathematics that was for me personally most difficult

V. Voevodsky

Univalent Foundations

Any mathematical object is an “object of” a category

Univalent Foundations

~~Any mathematical object is an "object of" a category~~

Any mathematical object is an object of a type

Univalent Foundations

~~Any mathematical object is an “object of” a category~~

Any mathematical object is an object of a type

Any type comes with its own notion of equality

Simple type theory

(3) Type theory: mostly unknown among mathematicians
(some exceptions: A. Turing, N.G. de Bruijn, T. Hales, V. Voevodsky)

$\rightarrow$ systems HOL, Isabelle, Idris, Coq, Agda

1908 Russell *Mathematical Logic as Based on the Theory of Types*

1940 Church *A Formulation of the Simple Theory of Types*

Simple type theory

Uses λ-calculus notation

Notation which is at the basis of functional programming (LISP,Scheme, Miranda, O'Caml, Haskell) and denotational semantics

Through the work of P.J. Landin, this calculus is very appropriate to describe many features of programming languages

Simple type theory

In set theory, a function is essentially a *functional graph* a "static" notion

Through λ-calculus, we can represent *functions* as *programs*

Simple types: a type o which represents the type of "propositions"

A type ι which represents the type of "individuals"

Function type: $\alpha \to \beta$, or $(\beta)\alpha$ in Church's notation

Such a type system is essentially the one used in O'Caml or Haskell

Addition of type variables; already suggested in R. Gandy's PhD thesis 1952

Simple type theory

Two notions of functions: *functional graph* or *program*

How to connect the two notions?

Any functional graph has to determine a program

Description operator $\iota x.\varphi$ and axiom

$(\exists ! x : \alpha)\varphi \ \rightarrow \ \varphi(\iota x.\varphi)$

We then have the "axiom of unique choice"

$(\forall x : \alpha)(\exists ! y : \beta)\psi(x, y) \ \rightarrow \ (\exists f : (\beta)\alpha)(\forall x : \alpha)\psi(x, f(x))$

To use $(\exists ! x : \alpha)\varphi$ assumes that we have a notion of *equality* on the type α

What is equality in mathematics?

In *set theory*, the axiom of extensionality states that two sets are equal if they have the same(!) elements

What is equality in mathematics?

In Church's system we have two form of the axiom of extensionality

10^o two equivalent propositions are equal

$$(\varphi \equiv \psi) \;\rightarrow\; \varphi = \psi$$

$10^{\alpha\beta}$ two pointwise equal functions are equal

$$((\forall x : \alpha)\; f(x) = g(x)) \;\rightarrow\; f = g$$

Church notices that 10^o allows to identify classes with propositional functions

What is equality in mathematics?

Equality $x = y$ is defined as

$(\forall P : \alpha \to o)\ P\ x \to P\ y$

$x = y$ is of type o if $x\ y$ are of type α

$\varphi \equiv \psi$ is defined to be $\varphi \to \psi\ \wedge\ \psi \to \varphi$

We can rewrite the extensionality axioms

$(\varphi = \psi) \equiv (\varphi \equiv \psi)$

$(f = g) \equiv ((\forall x : \alpha)\ f(x) = g(x))$

What is equality in mathematics?

In Church's system equality $x = y$ is defined as

$(\forall P\!:\!\alpha \rightarrow o)\ P\ x \rightarrow P\ y$

Henkin, and then P. Andrews, gave alternative presentations of type theory taking equality as primitives

Cf. Lambek and Scott "Introduction to Higher-Order Categorical Logic"

What is equality in mathematics?

One can *justify* these axioms by *defining* equality by induction on the types

In set theory, one can also interpret the extensionality axiom

Can we justify/explain the operation $\iota x.\varphi$?

How to “explain” logical laws

1908 Russell *Mathematical Logic as Based on the Theory of Types*

1940 Church *A Formulation of the Simple Theory of Types*

1973 Martin-Löf *An Intuitionistic Theory of Types: Predicative Part*

(Curry, de Bruijn, Howard, Tait, Scott, Martin-Löf, Girard, ...)

To consider *propositions* as *types*

How to “explain” logical laws

This “explains” some logical laws such as

modus-ponens: from $A \to B$ and A we can deduce B

introduction: if we can deduce B from A we can deduce $A \to B$

This reduces the problem of proof-checking to the problem of type-checking

Introduction of dependent type $B\ (x : A)$ and dependent product $(\Pi x : A)B$ and dependent sum $(\Sigma x : A)B$

Univalent foundation

In type theory and therefore in Univalent Foundations the fundamental notion is that of a dependent function, not of a "straight" function, and this is what makes the whole system successful

Type system

We can also introduce a notion of "disjoint sum"

The elements are $\mathsf{inl}(a)$ with a in A or $\mathsf{inr}(b)$ with b in B

We can define function by cases

We can also introduce data types like the type of natural numbers

This strengthens the connection with programming languages

constructors corresponds to *introduction rules*

Type of propositions

A proposition is a (special kind of) type

For the type of proposition we need a "type of types"

Universe U

What is the equality on U?

When are two types equal?

Rules for equality

P. Martin-Löf (1973) introduces a primitive notion of equality $\mathsf{Eq}_A(a_0, a_1)$

What are the laws?

Any element is equal to itself $1_a : \mathsf{Eq}_A(a, a)$

Leibnitz's law $C(a)$ implies $C(x)$ if $p : \mathsf{Eq}_A(a, x)$

Rules for equality

New law

$\mathsf{J}(d) : (\Pi x_0\ x_1 : A)(\Pi p : \mathsf{Eq}_A(x_0, x_1))C(x_0, x_1, p)$

given $d : (\Pi x : A)C(x, x, 1_x)$

Computation rule: $\mathsf{J}(d)\ x\ x\ 1_x = d\ x$

This generalizes

$(\forall x\ y)\mathsf{Eq}_A(x, y) \rightarrow C(x, y)$ given $(\forall x)C(x, x)$

Strengthening similar to the one for existence and disjunction

Rules for equality

Other formulation

$\mathsf{J}'(d) : (\Pi x : A)(\Pi p : \mathsf{Eq}_A(a, x))C(x, p)$

given $d : C(a, 1_a)$

Computation rule: $\mathsf{J}'(d)\ a\ 1_a = d$

This generalizes

$(\forall x)\mathsf{Eq}_A(a, x) \to C(x)$ given $C(a)$

Problem: are these two formulations equivalent?

The singleton law

A.k.a. the "vacuum cleaner power chord principle"

The type $S_a = (\Sigma x : A)\mathsf{Eq}_A(a, x)$ has for element (x, p) with $p : \mathsf{Eq}_A(a, x)$

In particular it has for element $(a, 1_a)$

Singleton Law $(\Pi z : S_a)\mathsf{Eq}_{S_a}((a, 1_a), z)$

This *follows* from the first formulation, and this *implies* the second formulation

(The fact that the two formulations are equivalent was considered to be a tricky result before work on univalent foundations)

New logical law for equality (unknown until 1973)!

The singleton law

"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."

Stratification of types

As soon as we introduce $\mathsf{Eq}_A(x, y)$ as a type we can iterate this operation

$\mathsf{Eq}_{\mathsf{Eq}_A(x,y)}(p, q)$

A type A is a *proposition*

$(\Pi x_0\ x_1 : A)\mathsf{Eq}_A(x_0, x_1)$

A type A is a *set*

$(\Pi x_0\ x_1 : A)\mathsf{prop}(\mathsf{Eq}_A(x_0, x_1))$

a.k.a $\mathsf{UIP}(A)$ "Uniqueness of Identity Proof"

Question (around 1991): is any type a set?

Stratification of types

It seems possible to show

$(\Pi\ x : A)(\Pi p : \mathsf{Eq}_A(x, x))\mathsf{Eq}(1_x, p)$

and it is instructive to understand where is the problem with this

This statement is actually *equivalent* to $\mathsf{UIP}(A)$

Stratification of types

N_1 is a proposition

N_0 is a proposition

Stratification of types

A is a groupoid

$(\Pi x_0\ x_1 : A)\mathsf{set}(\mathsf{Eq}_A(x_0, x_1))$

Theorem: *any type satisfies the groupoid laws*

Follows from the singleton law

Application: $\mathsf{UIP}(A)$ iff each $\mathsf{Eq}_A(a, a)$ has one element

Stratification of types

Theorem: *any function* $f : A \to B$ *satisfies the functorial laws*

Any $f : A \to B$ defines

$\mathsf{pMap}(f) : \mathsf{Eq}_A(a_0, a_1) \to \mathsf{Eq}_B(f\ a_0, f\ a_1)$

We can define 2-groupoids, 3-groupoids, $\ldots$

Type theory appears to be a generalization *of set theory*

Properties of Equality

$1_a : \mathsf{Eq}_A(a, a)$

$\mathsf{transp} : C(a) \to \mathsf{Eq}_A(a, x) \to C(x)$

$\mathsf{Eq}_{C(a)}(\mathsf{transp}\ u\ 1_a, u)$

$\mathsf{Eq}_{(\Sigma x:A)\mathsf{Eq}_A(a,x)}((a, 1_a), (x, p))$

Univalence Axiom

Constructive mathematics

Let N_k be the data type with k element

Let N be the data type of natural numbers

The type N_0 represents the empty type/the absurd proposition

Let $\neg A$ be the type $A \rightarrow N_0$

We say that A is "decidable" $A + \neg A$

A type A is *discrete* if each type $\mathsf{Eq}_A(x, y)$ is decidable

Proposition: *The types N_k and the type N are discrete*

Hedberg's Theorem

Theorem: (Hedberg, 1996) *Any discrete type is a set*

Lemma: *If we have an operation* $f\ p : \mathsf{Eq}_A(x,y)$ *for* $p : \mathsf{Eq}_A(x,y)$ *then* $\mathsf{Eq}(f\ p, (f\ 1)\cdot p)$ *for any* $p : \mathsf{Eq}_A(x,y)$

Lemma: *If* C *is decidable we can define* $f : C \rightarrow C$ *with a proof of* $(\Pi x\ y : C)\mathsf{Eq}_C(f\ x, f\ y)$

Given extensionality we can weaken the hypothesis to

$\neg\neg C \ \rightarrow \ C$

instead of C decidable

Constructive mathematics

N_2, N are sets

With one universe, we can show that they are *not* propositions

SSReflect considers only *decidable* structures because of this result

If A is a set we will see later that we have

$\mathsf{Eq}_{(\Sigma x:A)B}((a, b_0), (a, b_1)) \ \rightarrow \mathsf{Eq}_{B(a)}(b_0, b_1)$

but this implication *may fail* if A is not a set

Equivalence

Voevodsky gives a simple and uniform notion of equivalence between two types

If A and B are sets, this gives the notion of *bijection*

If A and B are propositions, this gives the notion of *logical equivalence*

If A and B are groupoids, this gives the notion of *categorical equivalence* of groupoids

Equivalence

If $f : A \to B$ the *fiber* of f at $b : B$ is the type

$F(b) = (\Sigma x : A)\mathsf{Eq}_B(b, f\ x)$

f is an equivalence iff this fiber is *contractible* at each point b

$\mathsf{prop}(F(b)) \times F(b)$

We write $A \simeq B$ for $(\Sigma f : A \to B)\mathsf{Equiv}(f)$

Proposition: *The identity function on any type is an equivalence*

This is not trivial!

Equivalence

If $g : C \to D$ is an equivalence we have a proof of $(\Pi d : D) F(d)$

Hence there exists a function $h : D \to C$ which is a *section* of g

$(\Pi x : D)\mathsf{Eq}_B(x, g\ (h\ x))$

Univalence Axiom

Since $X \simeq X$ we have a map $\mathsf{Eq}_U(A, B) \to A \simeq B$

The *Univalence Axiom* states that this map is an equivalence

In particular we have

$A \simeq B \to \mathsf{Eq}_U(A, B)$

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

Lemma: *If A and B are two propositions and $B \to A$ then any map $A \to B$ is an equivalence*

This will be proved later

Univalence Axiom

We have $\mathsf{Eq}_U(A, B)$ if A and B are equivalent propositions

We can hope closer connections to HOL

and to the SSReflect style of doing proofs (equational reasoning)

E.g. $\mathsf{Eq}_A(a_0, a_1) \rightarrow \mathsf{Eq}_A(u_0, u_1) \rightarrow \mathsf{Eq}_U(\mathsf{Eq}_A(a_0, u_0), \mathsf{Eq}_A(a_1, u_1))$

Univalence Axiom

The Univalence Axiom implies that two isomorphic sets are equal

It also implies that two equivalent groupoids are equal

Voevodsky has shown that this axiom implies *function extensionality*

From now on, we shall work with Univalence and hence function extensionality

Equivalence

We say that $g : B \to A$ is a *section* of $f : A \to B$ iff $\mathsf{Eq}(f \circ g, \mathsf{id}_B)$

We say that f is a *retraction* of g iff $\mathsf{Eq}(g \circ f, \mathsf{id}_A)$

f and g are inverse iff g is a section of f and f a retraction of g

Equivalence

Graduate Lemma: *If f and g are inverse then f is an equivalence*

Given this we can prove e.g.

Lemma: *If A and B are two propositions and $B \to A$ then any map $A \to B$ is an equivalence*

Equivalence

The proof of the Graduate Lemma consists itself in several Lemmas

Lemma 1: *If* $f : A \to B$ *has a left inverse* g *then* $\mathsf{prop}(B)$ *implies* $\mathsf{prop}(A)$

Lemma 2: *If* f *has a left inverse, then*

$$\mathsf{pMap}(f) : \mathsf{Eq}_A(a_0, a_1) \to \mathsf{Eq}_B(f\ a_0, f\ a_1)$$

has a left inverse

Lemma 3: *If we have a family of maps* $\varphi_x : D(x) \to E(x)$ *and for all* $x : A$*, the map* φ_x *has a left inverse then the map*

$$(\Sigma x : A)D(x) \to (\Sigma x : A)E(x)$$
$$(x, d) \longmapsto (x, \varphi_x\ d)$$

has a left inverse

Applications

Any involutive function is an equivalence

$\neg : N_2 \to N_2$ is an equivalence

We need function extensionality to prove $\neg \circ \neg = \mathsf{id}_{N_2}$

Hence we have a proof of $\mathsf{Eq}_U(N_2, N_2)$ which is not the reflexivity

By univalence $\mathsf{Eq}(\mathsf{Eq}_U(N_2, N_2), \mathsf{Equiv}(N_2, N_2))$

Hence U is not a set

It can be shown that U_1 is not a groupoid, U_2 is not a 2-groupoid and so on

Applications

Axiom of “choice”

$\mathsf{Eq}_U((\Pi x : A)(\Sigma y : B)C(x,y), (\Sigma f : A \to B)(\Pi x : A)C(x, f\ x))$

Distributivity law

$\mathsf{Eq}_U((\Sigma x : A)(B + C), (\Sigma x : A)B + (\Sigma x : A)C)$

Induction principle

$\mathsf{Eq}_U((\Pi z : A + B)C(z), (\Pi x : A)C(\mathsf{inl}(x)) \times (\Pi y : B)C(\mathsf{inr}(y)))$

Univalence Axiom

We have

$\mathsf{Eq}_U(A \times B, B \times A)$

$\mathsf{Eq}_U(A \times (B \times C), (A \times B) \times C)$

$\mathsf{Eq}_U(A \times N_1, A)$

This is less surprising than it seems however

$A \times B$ is *not* convertible to $B \times A$

Univalence Axiom

We have $\mathsf{Eq}_U(N, N + N_1)$

Notice that this is *not* true in set theory

0 belongs to N but not to $N + N_1$

a type structure is a "syntactical discipline for enforcing level of abstraction" Reynolds 1983

$0 : N$ is a *judgement* and not a *type* (using Martin-Löf terminology)

This can already be found in the design of Automath (1967)

Universal family of types

A family of types over A can be seen as a function $A \to U$

The map

$\mathsf{Eq}_U(A, B) \to A \simeq B$

gives for each $p : \mathsf{Eq}_U(A, B)$ a map

$\lambda x.\mathsf{transp}\ x\ p : A \to B$

and the transport of a family $C : A \to U$ can be seen as the composition of this map and $\mathsf{pMap}(C)$

Induction Principle for equivalences

If we have a statement $P(B, f)$ for

$B : U,\ f : A \to B,\ p : \mathsf{isEquiv}(A, B, f)$

For proving that $P(B, f)$ holds universally, it is enough to prove $P(A, \mathsf{id}_A)$

This follows from univalence and the singleton law

Application

E.g. define the type of pointed types $\mathsf{Pt} = (\Sigma X : U)X$

If $f : A \to B$ equivalence and $\mathsf{Eq}_B(f\ a, b)$ then

$\mathsf{Eq}_{\mathsf{Pt}}((A, a), (B, b))$

Application

Consider now the type $\mathsf{Fam} = (\Sigma X : U)X \to U$

This is the type of “families” (X, F) with $F : X \to U$

If $f : A \to B$ equivalence and $C : B \to U$ then

$\mathsf{Eq}_{\mathsf{Fam}}((A, C \circ f), (B, C))$

Hence

$\mathsf{Eq}_U((\Sigma x : A)C(f\ x), (\Sigma y : B)C(y))$

and

$\mathsf{Eq}_U((\Pi x : A)C(f\ x), (\Pi y : B)C(y))$

Back to the stratification

Using function extensionality we can show $\mathsf{prop}(\neg A)$ for *any* type A

Back to the stratification

Using function extensionality, we can prove

Theorem: *If we have* $(\Pi x : A)\mathsf{prop}(B)$ *then* $\mathsf{prop}((\Pi x : A)B)$

A can be any type in this statement

This is similar to impredicativity: any product of propositions is a proposition

What about a product of sets?

Back to the stratification

We have a notion of *property*: family of propositions over a type

To be an equivalence is a property of a function

The axiom of univalence is a proposition

Essential difference between *property* and *structure*

E.g. $f : A \to B$ isomorphism is a structure in general

and a property if A and B are sets

E.g. to be discrete is a property of a set, and hence a property of a type

Back to the stratification

If P is a proposition and $x\ y : P$, is $\mathsf{Eq}_P(x,y)$ a proposition?

Lemma: *If A is a type and we have $\varphi : (\Pi x : A)\mathsf{Eq}_A(a,x)$ then*

$$(\Pi x\ u : A)(\Pi p : \mathsf{Eq}_A(x,u))\mathsf{Eq}(p \cdot (\varphi\ u), \varphi\ x)$$

Direct from the singleton law and the groupoid laws

Back to the stratification

Corollary: *If* P *is a proposition and* $x\ y : P$ *then* $\mathsf{Eq}_P(x, y)$ *is contractible*

Q contractible means $\mathsf{prop}(Q) \times Q$, and this is a proposition

Back to the stratification

In particular any proposition is a set

Any set is a groupoid

Voevodsky's stratification is cumulative

Back to the stratification

To be a proposition, a set, $\ldots$ is a property

To prove

$(\Pi u\ v : \mathsf{prop}(A))\mathsf{Eq}(u, v)$

we use that $\mathsf{prop}(A) \rightarrow \mathsf{set}(A)$ and function extensionality

By function extensionality, $\mathsf{set}(A)$ is a proposition as well

Back to the stratification

N_0 is a proposition

N_1 is contractible

$N_2, N_3, \ldots$ and N are sets (by Hedberg's Theorem)

With a universe one can prove that they are *not* propositions

Equality in a sigma type

If $B(x)$ $(x : A)$ is a family of type and $p : \mathsf{Eq}_A(a_0, a_1)$ we can define

$B(a_0) \rightarrow B(a_1)$

$b \qquad \longmapsto \mathsf{transp}\ b\ p$

We have

$\mathsf{Eq}_U(\mathsf{Eq}_{(\Sigma x:A)B}((a_0, b_0), (a_1, b_1)), (\Sigma p : \mathsf{Eq}_A(a_0, a_1))\mathsf{Eq}_{B(a_1)}(\mathsf{transp}\ b_0\ p, b_1))$

Equality in a sigma type

We have

$\mathsf{prop}(A),\ (\Pi x : A)\mathsf{prop}(B) \to \mathsf{prop}((\Sigma x : A)B)$

$\mathsf{set}(A),\ (\Pi x : A)\mathsf{set}(B) \to \mathsf{set}((\Sigma x : A)B)$

In particular

$\mathsf{prop}(A),\ (A \to \mathsf{prop}(B)) \to \mathsf{prop}(A \times B)$

$\mathsf{prop}(A) \times A$ is a proposition

Equality in a sigma type

It follows that "to be contractible"

$\mathsf{prop}(A) \times A$

is a proposition

It then follows by extensionality that to be an equivalence is a property

We have $\mathsf{Eq}_U(\mathsf{prop}(A) \times A, \mathsf{Eq}_U(A, N_1))$

In particular A is contractible iff $\mathsf{Eq}_U(A, N_1)$

Equality in a sigma type

If A is a set and

$\mathsf{Eq}_{(\Sigma x:A)B(x)}((a,b_0),(a,b_1))$

then

$\mathsf{Eq}_{B(a)}(b_0,b_1)$

This is because $\mathsf{Eq}_A(a,a)$ is a proposition

Equality in a sigma type

We say that $g : C \to D$ is an embedding iff

$\mathsf{pMap}(g) : \mathsf{Eq}_C(x, y) \to \mathsf{Eq}_D(g\ x, g\ y)$

is an equivalence

This is a property of g

If C and D are sets, this is the same as being *injective*

$\mathsf{Eq}_D(g\ x, g\ y) \to \mathsf{Eq}_C(x, y)$

Equality in a sigma type

Proposition: *If $B(x)$ is a family of propositions over a type A then the first projection $(\Sigma x : A)B \ \rightarrow \ A$ is an embedding*

If A is a *set* we can think of $(\Sigma x : A)B$ as the

subset of elements in A satisfying B

Cf. definition of subset in Bishop's mathematics

Applications

Given A

There are two notions of family of types on A

$A \to U$ and $\mathsf{Slice}(A) = (\Sigma X : U)X \to A$

Any $F : A \to U$ defines a pair

$u(F) \ = \ (\Sigma x : A)F\ x, (\lambda z)z.1$

in $\mathsf{Slice}(A)$

Theorem: *The map* $u : (A \to U) \to \mathsf{Slice}(A)$ *is an equivalence*

Applications

If $A\ B : U$ we can define $T : N_2 \rightarrow U$ by

$T\ 0 = A$ $\qquad\qquad$ $T\ 1 = B$

Proposition: *We have* $\mathsf{Eq}_U(A + B, (\Sigma x : N_2)T\ x)$

This implies e.g. that $A + B$ is a set if A and B are sets

Product of sets?

Lemma: *The canonical map (and actually* any *map)*

$\mathsf{Eq}_{(\Pi x:A)B}(f, g) \to (\Pi x : A)\mathsf{Eq}_B(f\ x, g\ x)$

is an equivalence

The proof is incredible

Product of sets?

Let $C = (\Pi x : A)B$

We notice that both

$(\Sigma g : C)\mathsf{Eq}_C(f, g)$ and $(\Sigma g : C)(\Pi x : A)\mathsf{Eq}_B(f\ x, g\ x)$

are contractible

By "choice"

$\mathsf{Eq}_U((\Sigma g : C)(\Pi x : A)\mathsf{Eq}_B(f\ x, g\ x), (\Pi x : A)(\Sigma y : B)\mathsf{Eq}_B(f\ x, y))$

Corollary: *The product of* any *family of sets is a set*

Product of sets?

Lemma: *if we have a family of maps* $\varphi_x : D(x) \rightarrow E(x)$ *for* $x : C$ *and the total map* $(x, b) \longmapsto (x, \varphi_x\ b)$ *is an equivalence then each map* φ_x *is an equivalence*

This is a standard lemma in the theory of fibrations in homotopy theory

E.g. P. May *A concise course in algebraic topology*

Product of sets?

The proof is a sequence of equality, where $F = (\Sigma x : C)D$ and $G = (\Sigma x : C)E$

$(\Sigma(x, d) : F)\mathsf{Eq}((x_0, e_0), (x, \varphi_x\ d))$

$(\Sigma x : C)(\Sigma d : D(x))(\Sigma p : \mathsf{Eq}_C(x_0, x))\mathsf{Eq}_{E(x)}(\mathsf{transp}\ e_0\ p, \varphi_x\ d)$

$(\Sigma x : C)(\Sigma p : \mathsf{Eq}_C(x_0, x))(\Sigma d : D(x))\mathsf{Eq}_{E(x)}(\mathsf{transp}\ e_0\ p, \varphi_x\ d)$

$(\Sigma u : N_1)(\Sigma d : D(x_0))\mathsf{Eq}_{E(x_0)}(e_0, \varphi_{x_0}\ d)$

$(\Sigma d : D(x_0))\mathsf{Eq}_{E(x_0)}(e_0, \varphi_{x_0}\ d)$

Set mathematics

What is a poset?

It should be a *set* A

with a relation $R : A \to A \to U$ such that $\mathsf{prop}(R\ x\ y)$

this relation should be reflexive and transitive

and we have $\mathsf{Eq}_U(\mathsf{Eq}_A(x, y), R\ x\ y \times R\ y\ x)$

Set mathematics

What is a group?

It should be a *set* A

with some operations

$m : A \to A \to A \qquad inv : A \to A \qquad e : A$

and the usual axioms

We can form the type of all groups

Transport of structures

Let $\mathsf{Grp}(A)$ be the type of structures of groups over the set A

If $f : A \to B$ is a bijection between the sets A and B

then we have $\mathsf{Eq}_U(A, B)$ by univalence

Hence we have a map $\mathsf{Grp}(A) \to \mathsf{Grp}(B)$

Any group structure on A can be *transported* into a group structure on B along the bijection f

Transport of structures

On the other hand there is a direct notion of transport of operations

$m_B\ b\ b' = f\ (m_A\ (g\ b)\ (g\ b'))$ $\qquad inv_B\ b = f\ (inv_A\ (g\ b))$ $\qquad e_B = f\ e_A$

where g is *the* inverse of f

The two notions of transport coincide

In particular, we get in this way that B satisfies the axioms of group with operations m_B, inv_B, e_B

Structures

The collection of binary sequences form a set because we know what it means for two binary sequences to be equal. Given two groups, or sets, on the other hand, it is generally incorrect to ask if they are equal; the proper question is whether or not they are isomorphic

Structures

The collection of all groups

$\mathsf{Group} = (\Sigma X : U)\mathsf{set}(X) \times \mathsf{Grp}(X)$

form a *groupoid*

Theorem: *The map* $\mathsf{Eq}_{\mathsf{Group}}(g_0, g_1) \rightarrow \mathsf{Iso}(g_0, g_1)$ *is an equivalence*

In particular, two isomorphic groups are “equal”

Resizing rule

We add $A : U_0$ if A is a type and $\mathsf{prop}(A)$

With the resizing rule, the system becomes *impredicative*

Resizing rule

Voevodsky also requires that the collection of all propositions is a small type (which will be a small set)

For expressing this, we introduce a type $\Omega : U_0$ with the rules

$(A, p) : \Omega$ if A is a type and $p : \mathsf{prop}(A)$

If $X : \Omega$ then $X.1 : U_0$ and $X.2 : \mathsf{prop}(X.1)$

Theorem: Ω *is a set*

The collection of sets form a topos with Ω as the set of truth-values

Propositional truncation

Using the resizing rule, Voevodsky defines the operation

$\mathsf{inh}(A) = (\Pi X : U)\mathsf{prop}(X) \to (A \to X) \to X$

What is important is

(1) $\mathsf{inh}(A)$ is a *proposition*

(2) $A \to \mathsf{inh}(A)$

(3) $\mathsf{inh}(A) \to \mathsf{prop}(X) \to (A \to X) \to X$

Intuitively $\mathsf{inh}(A)$ is a proposition expressing that A is inhabited

If A is a proposition we have $\mathsf{Eq}_U(A, \mathsf{inh}(A))$

Constructive mathematics

We can define *new* connectives

$(\exists x : A)B$ defined as $\mathsf{inh}((\Sigma x : A)B)$

$A \vee B$ defined as $\mathsf{inh}(A + B)$

These connectives are similar to (but different from) the corresponding connectives in topos theory

Lemma: *If A and B are incompatible propositions then* $\mathsf{Eq}_U(A + B, A \vee B)$

In particular, $A + B$ is a proposition in this case

$(\Pi X : U)\mathsf{prop}(X) \rightarrow X + \neg X$ is a proposition

Unique Choice

We also have the principle of unique choice, for C, D sets

$$(\forall x : C)(\exists! y : D)\psi(x, y) \;\rightarrow\; (\Sigma g : C \rightarrow D)(\forall x : C)\psi(x, g\ x)$$

Without this principle, we would have *two* notions of functions

Function as λ-term or as functional relation

For A set and B propositions we have

$$(\exists! x : A)B \rightarrow (\Sigma x : A)B$$

This can be seen as a refinement of Church's description operator

Constructive mathematics

We can define the *image* of a map $f : A \to B$

This is determined by the *property* $(\exists x : A)\mathsf{Eq}_B(b, f\ x)$ on B

Compare with the fiber $(\Sigma x : A)\mathsf{Eq}_B(b, f\ x)$

We have a satisfactory correspondance between *subsets* and *properties*

Constructive mathematics

Difference between $\mathsf{inh}(A)$ and $\neg\neg A$

We have $\mathsf{inh}(A) \ \rightarrow \ \neg\neg A$

Let $P(n)$ be a family of *decidable* propositions over N

Proposition: $\mathsf{inh}((\Sigma n : N)P(n)) \rightarrow (\Sigma n : N)P(n)$

This is remarkable since $(\Sigma n : N)P(n)$ needs not be a proposition

Theorem: $\neg\neg((\Sigma n : N)P(n)) \rightarrow (\Sigma n : N)P(n)$ *is not provable*

Constructive mathematics

No reason any more why countable choice

$$(\forall n : N)(\exists x : X)R(n, x) \quad \rightarrow \quad (\exists f : N \rightarrow X)(\forall n : N)R(n, f\ n)$$

should hold

This is like in topos theory

Constructive mathematics

Definition of quotient by an equivalence relation

Given A a type and $R : A \to A \to U$ a proposition valued family of types which is an equivalence relation we define an *equivalence class* to be a proposition valued $P : A \to U$ such that

P is inhabited $(\exists x : A)P(x)$

$P\ x \to P\ y \to R\ x\ y$

$P\ x \to R\ x\ y \to P\ y$

Notice that from P a given equivalence class we *cannot* extract in general any element in A satisfying P

Univeral property

Let A/R the *set* of equivalence classes

We have a canonical map $s : A \to A/R$ and $R\ x_0\ x_1 \to \mathsf{Eq}_{A/R}(s\ x_0, s\ x_1)$

Let $f : A \to B$ a map into a *set* B such that $R\ x_0\ x_1 \to \mathsf{Eq}_B(f\ x_0, f\ x_1)$

There exists a unique map $g : A/R \to B$ such that $g \circ s = f$

And this despite the fact that we cannot extract in general an element in a given equivalence class!

More example in algebra

A *ring* R will be represented as a *set* with the usual structure

a divides b will be defined as *there exists* x such that $ax = b$

If a is *regular* i.e. $au = 0 \rightarrow u = 0$ then this x is *uniquely determined* and we have an explicit division operation

More example in algebra

An *ideal* P is a subset of R satisfying the usual laws

The ideal is *prime* iff $ab\epsilon P \rightarrow a\epsilon P \vee b\epsilon P$

Notice the use of $\vee$

We say that an ideal I is *finitely generated* iff *there exists* a finite list $x_1, \ldots, x_n$ which generates I

Notice that given I finitely generated we cannot in general extract an explicit list of generators

Lemma: *If* $I \cdot J \subseteq P$ *and* P *is prime then* $I \subseteq P$ or $J \subseteq P$

More example in algebra

In general, we can introduce the witness of an existential statement as long as we use this witness to build an object which *does not depend* on the exact choice of this witness

An example in analysis

If we define the type of real numbers R as a quotient of the set of Cauchy sequences of rationals

We can define $x\#y$ as meaning $(\exists r > 0) r \leqslant |x - y|$

We can define the inverse function $(\Pi x : R) x\#0 \rightarrow R$

This is because the inverse is *uniquely* determined

More example in algebra

Even if we start with a discrete structure, some natural operations build non necessarily discrete structure

E.g. localization of a ring for a multiplicative monoid

What was missing before was the insight that structures in algebra should be represented by *sets*

Finite sets

Category of finite sets $N_{k+1} = N_k + N_1$

We say that X is *finite* iff there *exists* k such that $\mathsf{Eq}_U(X, N_k)$

$(\exists k : N)\mathsf{Eq}_U(X, N_k)$ is a *proposition*

If X is finite, we can extract its cardinality but *not* the given equality proof $\mathsf{Eq}_U(X, N_k)$ (there are $k!$ such proofs)

We use that $\mathsf{Eq}_U(N_k, N_l)$ implies $\mathsf{Eq}_N(k, l)$

If X has k elements we have $\mathsf{inh}(\mathsf{Eq}_U(\mathsf{Eq}_U(X, X), N_{k!}))$

If X is finite then X is a discrete set

Torsors

We consider only $\mathbb{Z}$-torsors

A torsor is a set X with a $\mathbb{Z}$-action such that for any u in X the map $n \longmapsto u + n,\ \mathbb{Z} \to X$ is an equivalence

and $\mathsf{inh}(X)$

If X is a torsor we cannot in general exhibit one element of X

There is always the trivial torsor triv where $X = \mathbb{Z}$

Theorem: $\mathsf{Eq}_U(\mathsf{Eq}_{\mathsf{Torsor}}(\mathsf{triv}, \mathsf{triv}), \mathbb{Z})$

cf. D. Grayson Foundations.Ktheory

Torsors

If X is a torsor we have

$(\Pi u_0\ u_1 : X)(\exists! n : \mathbb{Z})\mathsf{Eq}_X(u_0 + n, u_1)$

and so, by unique choice we have an application

$X \times X \to \mathbb{Z}$

$(u_0, u_1) \longmapsto u_1 - u_0$

such that $\mathsf{Eq}_X(u_0 + u_1 - u_0, u_1)$

Resizing rule

What about a resizing rule for sets?

We *cannot* add $A : U_0$ if A is a type and $\mathsf{set}(A)$

We cannot have

$T = (\Pi X : U_0)\mathsf{set}(X) \to (((X \to \Omega) \to \Omega) \to X) \to X$

in U_0, since then $\mathsf{Eq}_{U_0}(T, (T \to \Omega) \to \Omega)$ which implies N_0

Cf. Reynolds non set theoretic model of polymorphisms

So we still need the hierarchy of universe

Global choice

We have

$\neg((\Pi X : U)\mathsf{set}(X) \to \mathsf{inh}(X) \to X)$

which is the negation of the axiom of global choice

No global choice function is invariant by isomorphism

Category theory

Look at the example of the collection of groups

A *category* is given by a type of objects A which should be a *groupoid*

We have $\mathsf{Hom}(a_0, a_1)$ which is a *set*

with the usual operations and laws of composition

We can then define the *set* of isomorphisms $\mathsf{Iso}_A(a_0, a_1)$

We have a map $\mathsf{Eq}_A(a_0, a_1) \to \mathsf{Iso}_A(a_0, a_1)$

This map should be an equivalence (a bijection in this case)

Category theory

Compare with the definition in set theory

Notion of “locally small” category

But this refers to the “size” and not to the complexity of equality

Category theory

We can define the notion of *equivalence* between two categories

We can consider the collection of "all" categories

Two equivalent categories are equal

B. Ahrens, C. Kapulkin, M. Shulman
"Univalent categories and the Rezk completion"

Category theory

The notion of Galois connections between posets

The notion of adjunction between categories

Category theory

Let $F : A \to B$ be a functor between two categories A and B

Like a monotone map at the level of groupoids

F is *essentially surjective* iff

$(\Pi b : B)(\exists a : A)\mathsf{Iso}_B(b, F\ a)$

This is a *property* of the functor

F is *full and faithful* iff the map

$\mathsf{Hom}_A(a_0, a_1) \to \mathsf{Hom}_B(F\ a_0, F\ a_1)$

is an equivalence (here a bijection)

Category theory

Lemma: *If F is full and faithful then for any b in B the type*

$$(\Sigma a : A)\mathsf{Iso}_B(b, F\ a)$$

is a proposition

Corollary: *If F is full and faithful and essentially surjective then F is an equivalence of categories*

We use the *principle of unique choice* at the level of groupoids!

Category theory

This definition of category solves some foundational issues that are somewhat disturbing when category theory is formulated in set theory

For instance, what should be a *cartesian* category (i.e. category with binary product)?

Should the product of two objects given explicitely as a function?

We have two notions (if we don't assume choice)

Category theory

In the univalent foundation, there is no problem since the product of two objects is uniquely determined up to isomorphism

And hence *up to equality* by definition of category

Since we have uniuqe choice, we have an explicit product function on objects

Beyond groupoids

U_0 is not a set

U_1 is not a groupoid

For 2-groupoids new phenomena arise

E.g. Eckmann-Hilton

$\mathsf{Eq}_{\mathsf{Eq}_{\mathsf{Eq}_A(x,x)}(1_x,1_x)}(\alpha \cdot \beta, \beta \cdot \alpha)$