Equality and dependent type theory

Oberwolfach, March 2 (with some later corrections)

The Axiom of Univalence, a type-theoretic view point

In type theory, we reduce proof-checking to type-checking

Hence we want type-checking to be decidable

This holds as soon as we have the normalization property

To add an axiom does not destroy the normalization property

A type-theoretic view point

Normalization property is obtained by giving a computational justification of each new construct

Another property that follows from this method is that we know that any closed term of type N reduces to a numeral, and any term of type $\sum_{x:A} B$ reduces to a pair

If we add an axiom, we destroy these properties

A type-theoretic view point

Project: to give a computational justification of the univalence axiom

We try to get this as a *model construction*

This involves two parts

Part 1: to give a purely axiomatic presentation of equality

Part 2: to give an interpretation of these axioms

This talk

We present various axiomatizations of the equality type (this is complete, and has been formally checked by Nils Anders Danielsson)

We then skecth a possible computational interpretation of these axioms (this has been checked in special cases only)

Part 1: Equality in type theory

First edition of *Principia Mathematica* (1910): no axiom of extensionality, but axiom of reducibility (propositions form a type, and we can quantify over any type, also known as *impredicativity*)

Second edition (1925): under the influence of Wittgenstein, Russell introduces the principle of extensionality

a function of propositions is always a truth function, and a function occurs only in a proposition through its values

and sees this as a (partial) replacement of the axiom of reducibility

Equality in type theory

A function can only appear in a matrix though its values

"This assumption is fundamental in the following theory. It has its difficulties, but for the moment, we ignore them. It takes the place (not quite adequatly) of the axiom of reducibility"

Church's formulation of type theory

Simplification of Russell's theory of types

A type of proposition o, a type of individuals and function type $A \to B$

For instance $o \to o$ is the type of the operation of negation

We have the usual connectives on propositions

$p \to q : o$ for the implication if $p\ q : o$

quantifiers at any type $\forall x : A.\varphi :\ o$ if $\varphi : o\ [x : A]$

Church's formulation

Uses λ-calculus to represent terms (implicit in *Principia Mathematica*)

If $f : A \to B$ and $a : A$ then $f\ a : B$ the application of the function f to the argument a

If $t : B\ [x : A]$ then $\lambda x.t : A \to B$

The terms of type o are the propositions

Usual connectives and (classical) logical rules

Equality in Church's formulation

We can *define* an equality (Leibnitz equality) $\mathsf{Id}_A\ a_0\ a_1$ as

$\forall P : A \to o.\ \ P(a_0) \to P(a_1)$

This definition is *impredicative*

One can show that this is a reflexive, symmetric and transitive relation

The *axiom of extensionality* has then two forms

on propositions: $(p \leftrightarrow q) \to \mathsf{Id}_o\ p\ q$

on functions: $(\forall x : A.\ \mathsf{Id}_B\ (f\ x)\ (g\ x)) \to \mathsf{Id}_{A\to B}\ f\ g$

Equality in Church's formulation

Axiomatic presentation

$\mathsf{ax}_1 : \forall x : A.\ \mathsf{Id}_A\ x\ x$

$\mathsf{ax}_2 : \mathsf{Id}_A\ a_0\ a_1 \to P(a_0) \to P(a_1)$

$\mathsf{ax}_3 : (p \leftrightarrow q) \to \mathsf{Id}_o\ p\ q$

$\mathsf{ax}_4 : (\forall x : A.\ \mathsf{Id}_B\ (f\ x)\ (g\ x)) \to \mathsf{Id}_{A \to B}\ f\ g$

Dependent Type Theory

Curry-Howard, N. de Bruijn, D. Scott, P. Martin-Löf

Add to simple type theory the notion of *dependent type* $B(x)$ type for $x : A$

$\prod_{x:A} B(x)$ type of functions/sections f with $f\ a : B(a)$ if $a : A$

$\sum_{x:A} B(x)$ type of pairs a, b with $a : A$ and $b : B(a)$

Natural set theoretic interpretation

Proposition as Types

If $B(x) = B$ does not depend on $x : A$

$\prod_{x:A} B(x)$ is written $A \to B$ represents both function type and implication

$\sum_{x:A} B(x)$ is written $A \times B$ represents both cartesian product and conjunction

Proposition as Types

$\prod_{x:A} B(x)$ represents

-universal quantification and

-the set of sections of the family $B(x)$

Proposition as Types

$\sum_{x:A} B(x)$ represents

-the fiber space over A defined by the family $B(x)$ and

-the set $\{x : A \mid B(x)\}$ and

-existential quantification $(\exists x : A)B(x)$

Universe

Martin-Löf (1972) introduces the notion of universe U, type of "small" types

U can be thought of both as a type of types and as a type of propositions

Predicative system

$\sum_{X:U} X \times (X \to X)$ or $\prod_{X:U} (X \to X)$ are large types and not of type U

$\sum_{X:U} X \times (X \to X)$

type of all structures with one constant and one unary operation

Some Notations

$A \to B \to C$ for $A \to (B \to C)$

$\lambda x\ y\ z.t$ for $\lambda x \lambda y \lambda z.t$

$\prod_{x_0\ x_1:A} B(x_0, x_1)$ for $\prod_{x_0:A} \prod_{x_1:A} B(x_0, x_1)$

Dependent Type Theory

To summarize: extension of Gödel's system T with

$\prod_{x:A} B(x)$ and $\sum_{x:A} B(x)$

A type of small types U (closed under products and sums)

$N_0,\ N_1,\ N_2,\ N : U$

Terms: λ-terms extended with constants $0 : N$ and $x + 1 : N\ [x : N]$ and

natrec $: P(0) \to (\prod_{x:N} P(x) \to P(x+1)) \to \prod_{x:N} P(x)$

natrec $a\ f\ 0 = a$ and natrec $a\ f\ (n+1) = f\ n$ (natrec $a\ f\ n$)

Dependent Type Theory

Uniform foundation for logic and type theory: True = Provable = Inhabited

(In Church's type theory, one needs to add logical rules to the type structure)

For instance

$$\prod_{A\ B:U} (A \to B \to A)$$

is true because it is inhabited by $\lambda A\ B\ x\ y.\ x$

$$A : U,\ B : U \vdash \lambda x\ y.\ x : A \to B \to A$$

$$A : U,\ B : U,\ x : A,\ y : B \vdash x : A$$

Inductive definitions

$N,\ N_0,\ N_1,\ N_2$

$W\ A\ B$ well-founded tree types

We work in the fragment of type theory with no identity type

Equality in Dependent Type Theory

We follow an $axiomatic$ approach: what should be the property of equality?

We should have a $type$ of equality proofs $\mathsf{Id}_A\ a_0\ a_1$ if A type and $a_0\ a_1 : A$

We write $\alpha, \beta, \ldots$ equality proofs

Some axioms

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$ given $B(x)$ type over $x : A$

We have $b \cdot \alpha : B(a_1)$ if $b : B(a_0)$ and $\alpha : \mathsf{Id}_A\ a_0\ a_1$

Equality as Path

We think of a type A as a $space$

A proof $\alpha : \mathsf{Id}_A\ a_0\ a_1$ is thought of as a $path$ between a_0 and a_1

The operation $b \cdot \alpha : B(a_1)$ for $b : B(a_0)$ corresponds then to the *path lifting property*

(For a covering space, this lifting property provides a bijection between two fibers of two connected points)

We expect to have $\mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

Equality as Path

3 axioms so far

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$

$\mathsf{ax}_3 : \mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

Contractible Spaces

If A is a type we define a new type iscontr A to be $\sum_{a:A} \prod_{x:A} \mathsf{Id}_A\ a\ x$

This means that A has *exactly one element*

In term of space, A is *contractible*

The justification of this last point is subtle: iscontr A seems at first to only say that A is inhabited and (path) connected

A further axiom

(J.P.Serre) *when I was working on homotopy groups (around 1950), I convinced myself that, for a space X, there should exist a fibre space E, with base X, which is contractible; such a space would allow me (using Leray's methods) to do lots of computations on homotopy groups... But how to find it? It took me several weeks (a very long time, at the age I was then) to realize that the space of "paths" on X had all the necessary properties-if only I dared call it a "fiber space". This was the starting point of the loop space method in algebraic topology.*

(Interview in the Matematical Intelligencer, 1986)

A further axiom

Given a point a in X, J.P. Serre was considering the space E of paths α from a to another point x of A, with the map $E \to A,\ \alpha \longmapsto x$

E is contractible, and we have a contractible fibre space E with base X

In type theory, this translates to

For $a : X$, the type $E = \sum_{x:X} \mathsf{Id}_A\ a\ x$ should be contractible

Any element $(x, \alpha) : E$ is equal to $(a, 1_a)$

Equality as Path

4 axioms

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$

$\mathsf{ax}_3 : \mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

$\mathsf{ax}_4 : \mathsf{iscontr}\ (\sum_{x:A} \mathsf{Id}_A\ a\ x)$

Equivalent formulation

introduction rule $1_a : \mathsf{Id}_A\ a\ a$

elimination rule: given $C(x, \alpha)$ for $x : A$ and $\alpha : \mathsf{Id}_A\ a\ x$ then we have

$$\mathsf{elim} : C(a, 1_a) \to \prod_{x:A} \prod_{\alpha:\mathsf{Id}_A\ a\ x} C(x, \alpha)$$

(C. Paulin's formulation of equality in type theory)

"computation" rule: $\mathsf{Id}_{C(a,1_a)}\ (\mathsf{elim}\ c\ a\ 1_a)\ c$ for any $c : C(a, 1_a)$

Dependent type version of $\mathsf{Id}_A\ a\ x \to P(a) \to P(x)$

Equivalent formulation

introduction rule $1_a : \mathsf{Id}_A\ a\ a$

elimination rule: given $C(x_0, x_1, \alpha)$ for $x_0\ x_1 : A$ and $\alpha : \mathsf{Id}_A\ x_0\ x_1$ we have

$$\mathsf{J} : (\prod_{x:A} C(x, x, 1_x)) \to \prod_{x_0\ x_1:A} \prod_{\alpha:\mathsf{Id}_A\ x_0\ x_1} C(x_0, x_1, \alpha)$$

"computation" rule: $\mathsf{Id}_{C(x,x,1_x)}\ (\mathsf{J}\ d\ x\ x\ 1_x)\ (d\ x)$ for any $d : \prod_{x:A} C(x, x, 1_x)$

This is P. Martin-Löf's formulation of equality in type theory

It expresses in type theory that Id_A is the least reflexive relation on A

Consequences of these axioms

All these different formulations are equivalent axiom systems (proved formally in type theory)

Given these axioms any type has automatically a *groupoid structure*

Proofs-as-programs version of the fact that equality is symmetric and transitive

Any function $f : A \to B$ defines a functor

Hofmann-Streicher 1992

Equality as Path

Most topological intuitions have a direct formal expression in type theory, e.g.

for any type X and $a : X$ the loop space $\Omega_1(X, a) = \mathsf{Id}_X\ a\ a$ has a group structure

$\Omega_2(X, a) = \Omega_1(\mathsf{Id}_X\ a\ a, 1_a), \ldots$

Equality as Path

We have (proved formally)

Proposition: (Čech, 1932) $\Omega_n(X, a)$ *is commutative for* $n \geqslant 2$

This is a corollary of the following fact.

Proposition: *If* X *with a binary operation and an element* $e : X$ *which is both a left and right unit for this operation then the group* $\Omega_1(X, e) = \mathsf{Id}_X\ e\ e$ *is commutative*

Equality as Path

Warning! Our statement is actually different from the usual statement

$\Omega_1(X, a)$ is defined as a space, which may have a complex equality

To get the usual statement, we would have to consider the *set* (as defined later) $\pi_1(X, x)$ associated to it

Axiom of extensionality

The usual formulation of this axiom is, with $F = \prod_{x:A} B(x)$

$$(\prod_{x:A} \mathsf{Id}_{B(x)}\ (f\ x)\ (g\ x))) \to \mathsf{Id}_F\ f\ g$$

(V. Voevodsky) This is equivalent to

A product of contractible types is contractible

$$(\prod_{x:A} \mathsf{iscontr}\ (B(x))) \to \mathsf{iscontr}\ (\prod_{x:A} B(x))$$

Equality as Path

5 axioms

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$

$\mathsf{ax}_3 : \mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

$\mathsf{ax}_4 : \mathsf{iscontr}\ (\sum_{x:A} \mathsf{Id}_A\ a\ x)$

$\mathsf{ax}_5 : (\prod_{x:A} \mathsf{iscontr}\ (B(x))) \to \mathsf{iscontr}\ (\prod_{x:A} B(x))$

Stratification of types

A is of h-level 0 iff A is contractible

A is of h-level 1 iff $\mathsf{Id}_A\ a_0\ a_1$ is contractible for any $a_0\ a_1 : A$

A is a $\mathit{proposition}$ iff A is of h-level 1

A is of h-level 2 iff $\mathsf{Id}_A\ a_0\ a_1$ is a proposition for any $a_0\ a_1 : A$

A is a set iff A is of h-level 2

$\ldots$

Stratification of types

These definitions can be internalised in type theory

$$\mathsf{isprop}\ A = \prod_{x_0\ x_1 : A} \mathsf{iscontr}\ (\mathsf{Id}_A\ x_0\ x_1)$$

$$\mathsf{isset}\ A = \prod_{x_0\ x_1 : A} \mathsf{isprop}\ (\mathsf{Id}_A\ x_0\ x_1)$$

There is no "global" type of all propositions like in an impredicative framework or a type of all sets

Extensionality and impredicativity

The extensionality axiom implies

-a product of propositions is always a proposition

$$\prod_{x:A} \text{isprop}\ (B(x)) \quad \rightarrow \text{isprop}\ (\prod_{x:A} B(x))$$

-a product of sets is always a set

$$\prod_{x:A} \text{isset}\ (B(x)) \quad \rightarrow \text{isset}\ (\prod_{x:A} B(x))$$

The first implication confirms Russell's remark that the principle of extensionality can replace in some cases the axiom of reducibility

Propositions

If we have isprop $(B(x))$ for all $x : A$ then the canonical projection

$$(\sum_{x:A} B(x)) \to A$$

is a mono, and we can think of $\sum_{x:A} B(x)$ as the $subset$ of elements in A satisfying the property $B(x)$

Unique Existence

iscontr$(\sum_{x:A} B(x))$ a generalisation of unique existence $\exists! x : A.B(x)$

If $B(x)$ is a proposition, iscontr$(\sum_{x:A} B(x))$ reduces to unique existence on x

More refined in general than to state that only one element in A satisfies $B(x)$

We always have iscontr$(\sum_{x:A} \mathsf{Id}_A\ a\ x)$ but $\mathsf{Id}_A\ a\ x$ may not be a proposition

Hedberg's Theorem

Define isdec A to be $\prod_{x_0\ x_1:A} \mathsf{Id}_A\ x_0\ x_1 + \neg\ (\mathsf{Id}_A\ x_0\ x_1)$

$\neg\ C$ denotes $C \to N_0$, where N_0 is the empty type

M. Hedberg noticed (1995) that we have

isdec $A \to$ isset A

In particular N the type of natural numbers is decidable

So N is a set but it is not a $proposition$ (since $\neg\ (\mathsf{Id}_N\ 0\ 1)$ is inhabited)

Other properties

isprop N_0, iscontr N_1, isset N_2

$\neg A \to$ isprop A

isprop (iscontr A) for all type A

isprop (isprop A) for all type A

isprop (isset A) for all type A

isprop A iff $\prod_{x_0\ x_1:A}$ iscontr($\mathsf{Id}_A\ x_0\ x_1$) iff $\prod_{x_0\ x_1:A}$ $\mathsf{Id}_A\ x_0\ x_1$

Axiom of extensionality

In Church's type theory $(p \leftrightarrow q) \to \mathsf{Id}_o\ p\ q$

What about adding as an axiom $(X \leftrightarrow Y) \to \mathsf{Id}_U\ X\ Y$?

S. Berardi noticed that this is contradictory (with dependent type theory):

If X inhabited X is logically equivalent to $X \to X$

We would have $\mathsf{Id}_U\ X\ (X \to X)$ and then X and $X \to X$ are isomorphic

X model of λ-calculus, hence any map on X has a fixed-point

and we get a contradiction if $X = N$ or $X = N_2$

Axiom of extensionality

In ordinary type theory, one can notice directly that if X is inhabited then X is logically equivalent to N_1 and hence X is a singleton

Axiom of extensionality

So we need a more subtle formulation

Define $\mathsf{Isom}\ X\ Y$ to be

$$\sum_{f:X\to Y}\ \sum_{g:Y\to X}\ (\prod_{x:X} \mathsf{Id}_X\ (g\ (f\ x))\ x) \times (\prod_{y:Y} \mathsf{Id}_Y\ (f\ (g\ y))\ y)$$

Extensionality axiom for small types (Hofmann-Streicher 1996)

$$\mathsf{Isom}\ X\ Y\ \to\ \ \mathsf{Id}_U\ X\ Y$$

Other properties

A consequence of this axiom is

$\neg(\mathsf{isset}\ U)$

Indeed, $\mathsf{Id}_U\ N_2\ N_2$ has two distinct elements

We have

If $\mathsf{isset}\ A$ and $\prod_{x:A} \mathsf{isset}\ (B(x))$ then $\mathsf{isset}\ (\sum_{x:A} B(x))$

$\mathsf{isset}\ A$ is not connected to the size of A but with the complexity of the equality on A

Equality as Path

6 axioms

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$

$\mathsf{ax}_3 : \mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

$\mathsf{ax}_4 : \mathsf{iscontr}\ (\sum_{x:A} \mathsf{Id}_A\ a\ x)$

$\mathsf{ax}_5 : (\prod_{x:A} \mathsf{iscontr}\ (B(x))) \to \mathsf{iscontr}\ (\prod_{x:A} B(x))$

$\mathsf{ax}_6 : \mathsf{Isom}\ X\ Y \to \mathsf{Id}_U\ X\ Y$

Univalence Axiom

For $f : Y \to X$ and $x_0 : X$, the *fiber* of f above x_0 is

$$f^{-1}(x_0) =_{def} \sum_{y:Y} \mathsf{Id}_X\ x_0\ (f\ y)$$

$\sum_{x:X} f^{-1}(x) = \sum_{x:X} \sum_{y:Y} \mathsf{Id}_X\ x\ (f\ y)$ is the graph of f

Any map $f : Y \to X$ *is isomorphic to a fibration* $(\sum_{x:X} f^{-1}(x)) \to X$

Univalence Axiom

We define what should be a "path" between two types X and Y

If $f : X \to Y$ we define when f is a $weak\ equivalence$

$\mathsf{isweq}\ f\ =_{def}\ \prod_{y:Y} \mathsf{iscontr}\ (f^{-1}(y))$

Theorem: *To be a weak equivalence is always a proposition, i.e.* $\mathsf{isprop}\ (\mathsf{isweq}\ f)$

We define $\mathsf{Weq}\ X\ Y$ to be $\sum_{f:X\to Y} \mathsf{isweq}\ f$

Univalence Axiom

Let isiso f be

$$\sum_{g:Y\to X} \left(\prod_{x:X} \mathsf{Id}_X\ (g\ (f\ x))\ x\right) \times \left(\prod_{y:Y} \mathsf{Id}_Y\ (f\ (g\ y))\ y\right)$$

isiso $f \leftrightarrow$ isweq f

However isweq f is a always a proposition while

isiso f may not be a proposition in general

Univalence Axiom

Warning! Weak equivalence is $stronger$ than logical equivalence, e.g.

$$\prod_{x:A} \sum_{y:B} R(x,y) \text{ and } \sum_{f:A\to B} \prod_{x:A} R(x, f\ x)$$

are weakly equivalent, since they are isomorphic

This is more precise than only to state logical equivalence

Univalence Axiom

Clearly we have Weq X X, because the identity map is a weak equivalence

Hence we have a map

$\mathsf{Id}_U\ X\ Y \to \mathsf{Weq}\ X\ Y$

The $Univalence\ Axiom$ states that this map is a weak equivalence

V. Voevodsky has shown that this implies functional extensionality

This axiom does not hold for the set-theoretic interpretation of type theory

Equality as Path

6 axioms

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

$(\cdot) : B(a_0) \to \mathsf{Id}_A\ a_0\ a_1 \to B(a_1)$

$\mathsf{ax}_3 : \mathsf{Id}_{B(a_0)}\ (b \cdot 1_{a_0})\ b$

$\mathsf{ax}_4 : \mathsf{iscontr}\ (\sum_{x:A} \mathsf{Id}_A\ a\ x)$

$\mathsf{ax}_5 : (\prod_{x:A} \mathsf{iscontr}\ (B(x))) \to \mathsf{iscontr}\ (\prod_{x:A} B(x))$

ax_6 : *The canonical map* $\mathsf{Id}_U\ X\ Y \to \mathsf{Weq}\ X\ Y$ *is a weak equivalence*

Invariance under isomorphisms

We get a formalism where two $isomorphic$ mathematical structures are $equal$

For instance on the type $S = \sum_{X:U} X \times (X \to X)$ we have (proved formally)

$\mathsf{Id}_S\ (X, a, f)\ (Y, b, g)$ iff the structures (X, a, f) and (Y, b, g) are isomorphic

This invariance property does not hold for set theory

Is this theory *consistent*?

Model

Since the paper

D. Kan *A combinatorial definition of homotopy groups*, Annals of Mathematics, 1958, 67, 282-312

a way to represent spaces is to use (Kan) simplicial sets

This model satisfies (and suggested?) the univalence axiom

Part 2: Computational interpretation

We have listed axiomatically some properties that the equality should have

All other notions in type theory are motivated/justified by computation rules

For instance

$$\text{natrec} : P(0) \to (\prod_{x:N} P(x) \to P(x+1)) \to \prod_{x:N} P(x)$$

is justified by $\text{natrec}\ a\ f\ 0 = a$ and $\text{natrec}\ a\ f\ (n+1) = f\ n\ (\text{natrec}\ a\ f\ n)$

(This represents at the same time both induction and recursion)

Can we justify in a similar way these axioms for equality?

Gandy's interpretation

On the Axiom of Extensionality

R. Gandy, The Journal of Symbolic Logic, 1956

Interpret $extensional$ type theory in $intensional$ type theory

The intuition is precisely that in λ-calculus a function can only occur in a proposition through its values in a term (cf. Russell's formulation of the axiom of extensionality)

This is only valid for $closed$ λ-terms: if X is a functional variable f does not appear in $X\ f$ through its values

Gandy's interpretation

The second part of the paper shows that a similar interpretation works for set theory

The paper is one of the first instance of the *logical relation* technique

We need to extend this technique to dependent types

Gandy's interpretation

Our current work is to adapt Gandy's interpretation to dependent types

Intuitively: we know what the equality should be on all base types (on the universe U it should be weak equivalence) and so we can define equality on each type by induction on the types

This is similar to the work on *observational type theory* (Thorsten Altenkirch, C. McBride) and on *two-level type theory* (M. Maietti, G, Sambin) but generalizes them to the case of computationally relevant identity proofs

This was also suggested by D. Turner (1989) for functional equality

Interpretation of equality

At type $N_0,\ N_1$ we define $\mathsf{Id}_{N_1}\ x\ y = N_1$

At type N we define $\mathsf{Id}_N\ 0\ 0 = N_1$ and $\mathsf{Id}_N\ (x+1)\ 0 = \mathsf{Id}_N\ 0\ (y+1) = N_0$ and $\mathsf{Id}_N\ (x+1)\ (y+1) = \mathsf{Id}_N\ x\ y$

For universe, we should say that $\mathsf{Id}_U\ A_0\ A_1$ is $\mathsf{Weq}\ A_0\ A_1$

Interpretation of equality

For sum types, we have $A : U$ and $F : A \to U$ and we can define if $S = \Sigma\ A\ F$

$$\mathsf{Id}_S\ (a_0, b_0)\ (a_1, b_1) = \sum_{\alpha : \mathsf{Id}_A\ a_0\ a_1} \mathsf{Id}_F\ a_1\ (F(\alpha)\ b_0)\ b_1$$

For product types, if $P = \Pi\ A\ F$

$$\mathsf{Id}_P\ f_0\ f_1 = \prod_{x:A} \mathsf{Id}_F\ x\ (f_0\ x)\ (f_1\ x)$$

So we have a recursive structure

Interpretation of equality

For instance the fact that all singleton types $\sum_{x:A} \mathsf{Id}_A\ a\ x$ are contractible can be checked by induction on A

The problem is to build a model of type theory, and the main problem is to validate the rule

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

Logical relation (Gandy)

A model of type theory where an element $a : A$ is interpreted by $a_0\ a_1 : A$ and a proof that a_0 and a_1 are related

The relation is defined by induction on A: for $A = o$ the relation is logical equivalence and for function types $A \to B$ we have $\mathsf{Id}_{A\to B}\ f_0\ f_1$ iff

$$\mathsf{Id}_A\ a_0\ a_1 \to \mathsf{Id}_B\ (f_0\ a_0)\ (f_1\ a_1)$$

Logical relation with dependent types

We define $\mathsf{EQ}\ A_0\ A_1$ and if $\alpha : \mathsf{EQ}\ A_0\ A_1$ a relation $\mathsf{EQ}\ {}_\alpha\ a_0\ a_1$ for $a_0 : A_0$ and $a_1 : A_1$

This relation is defined in an inductive-recursive way

We define by recursion $A' : \mathsf{EQ}\ A\ A$ and Id_A is the relation $\mathsf{EQ}_{A'}$

The base case is that any map $\alpha : A_0 \to A_1$ which is a weak equivalence determine a proof of $\mathsf{EQ}\ A_0\ A_1$ and $\mathsf{EQ}_\alpha\ a_0\ a_1$ is then $\mathsf{Id}_{A_1}\ (\alpha\ a_0)\ a_1$

Logical relation with dependent types

If we have α : EQ A_0 A_1 and $\beta(\omega)$: EQ $(F_0\ a_0)$ $(F_1\ a_1)$ we introduce $\Sigma\ \alpha\ \beta$: EQ S_0 S_1 where $S_i = \Sigma\ A_i\ F_i$ and

$$\mathsf{EQ}_{\Sigma\ \alpha\ \beta}\ (a_0, b_0)\ (a_1, b_1) = (\Sigma\ \omega : \mathsf{EQ}_\alpha\ a_0\ a_1)\ \mathsf{EQ}_{\beta(\omega)}\ b_0\ b_1$$

Logical relation with dependent types

If we have α : EQ A_0 A_1 and $\beta(\omega)$: EQ $(F_0\ a_0)$ $(F_1\ a_1)$ we introduce $\Pi\ \alpha\ \beta$: EQ P_0 P_1 where $P_i = \Sigma\ A_i\ F_i$ and

$$\mathsf{EQ}_{\Pi\ \alpha\ \beta}\ f_0\ f_1 = (\Pi\ \omega : \mathsf{EQ}_\alpha\ a_0\ a_1)\ \mathsf{EQ}_{\beta(\omega)}\ (f_0\ a_0)\ (f_1\ a_1)$$

Logical relation with dependent types

In a way similar to dependent sums we define $\mathsf{Id}_\Gamma\ \sigma_0\ \sigma_1$ by

$\mathsf{Id}_{\Gamma.A}\ (\sigma_0, a_0)\ (\sigma_1, a_1) = (\Sigma\ \alpha : \mathsf{Id}_\Gamma\ \sigma_0\ \sigma_1)\mathsf{EQ}_{A\alpha}\ a_0\ a_1$

and we have if $\Gamma \vdash A$ and $\alpha : \mathsf{Id}_\Gamma\ \sigma_0\ \sigma_1$ then $A\alpha : \mathsf{EQ}\ A\sigma_0\ A\sigma_1$

If $\Gamma \vdash t : A$ we have $t\alpha : \mathsf{EQ}_{A\alpha}\ t\sigma_0\ t\sigma_1$

What is important is that we have $A\alpha = B\alpha$ if $\Gamma \vdash A = B$

Logical relation with dependent types

Using such a logical relation we solve the problem with the conversion rule

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

Also, any $f : A_0 \to A_1$ which is a weak equivalence (which has an homotopic inverse) gives a proof of EQ A_0 A_1; this is the base case of the inductively defined relation EQ

Logical relation with dependent types

What is lacking at this point is the converse: to any proof $\alpha : \mathsf{EQ}\ A_0\ A_1$ should correspond two maps $\alpha^+ : A_0 \to A_1$ and $\alpha^- : A_1 \to A_0$ such that $\mathsf{EQ}_\alpha\ a_0\ a_1$ is equivalent to $\mathsf{Id}_{A_1}\ (\alpha^+\ a_0)\ a_1$ and equivalent to $\mathsf{Id}_{A_0}\ a_0\ (\alpha^-\ a_1)$

Logical relation with dependent types

Some special case

For instance the case where U contains $N,\ N_0,\ N_1,\ N_2$ and is closed only by $+$ and $\to$

If A_0 and A_1 are in U we can define directly $\alpha^+,\ \alpha^-$ for any $\alpha : \mathsf{EQ}\ A_0\ A_1$ by induction on α

Using this technique, it can be shown that any term $F : U \to U$ defines a functor

Logical relation with dependent types

We can then try to analyze the equality on types such as $\sum_{X:U} X \times (X \to X)$ and $\prod_{X:U} (X \to X)$

Implementation

Nils Anders Danielsson has proved formally that most properties proved by V. Voevodsky can be proved from a purely axiomatic presentation (no new computational rules)

This fact has been used crucially in this presentation

See www.cse.chalmers.se/~nad/listings/equality/README.html