Homotopy Type Theory

Steve Awodey

Philosophy Department Colloquium
Carnegie Mellon University
September 2023

Über die mathematische Logik

Von

Th. Skolem.

(Nach einem Vortrag gehalten im Norwegischen Mathematischen Verein am 22. Oktober 1928).

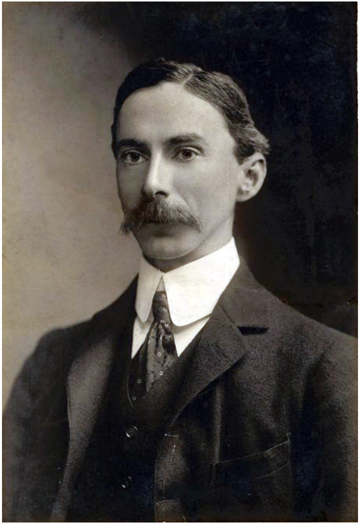
Bekanntlich wurde die Logik als Wissenschaft von Aristoteles gegründet. Alle kennen den Aristotelischen Syllogismus. Die syllogistischen Figuren von Aristoteles bildeten den Hauptinhalt der Logik während des ganzen Mittelalters. Kant soll einmal gesagt haben, dass die Logik die einzige Wissenschaft sei, die seit dem Altertum gar keine Fortschritte gemacht hatte. Dies war vielleicht damals richtig; heute ist es aber nicht mehr so.

In der letzten Zeit ist nämlich der sogenannte Logikkalkul oder die mathematische Logik entwickelt worden, eine Theorie, welche über die Aristotelische Logik weit hinausgekommen ist. Sie ist von Mathematikern entwickelt worden; die Philosophen von Fach haben sich sehr wenig dafür interessiert, vermutlich weil sie diese Theorie zu mathematisch gefunden haben. Anderseits haben auch die meisten Mathematiker sich sehr wenig dafür interessiert, weil sie die Theorie zu philosophisch gefunden haben.

Overview

- Homotopy Type Theory is a new branch of mathematical logic based on a recently discovered connection between topology (Homotopy theory) and logic (Type Theory).
- Martin-Löf type theory (a formal system of constructive foundations) can be interpreted into abstract homotopy theory (the mathematics of continuous space).
- Computerized proof systems based on MLTT can then be used to formalize higher mathematical reasoning.
- There are also conceptual reasons why this is a good foundation for modern mathematics.

Type Theory



Naive Type Theory (Frege)

Frege begins his *Basic Laws of Arithmetic* with a strict hierarchy:

- *Objects* stand opposed to *functions* ... I count as an object everything that is not a function
- *Functions of two arguments* are just as fundamentally distinct from *functions of one argument* as the latter are from *objects*.
- Functions whose arguments are objects we now call *first-level functions*, ... those whose arguments are first-level functions will be called *second-level functions*.

But he soon breaks these rules with his *value-ranges*:

- value-ranges seem to me one of the most consequential additions to my *Begriffsschrift* ... the domain of what can occur as an argument of a function is thereby extended.

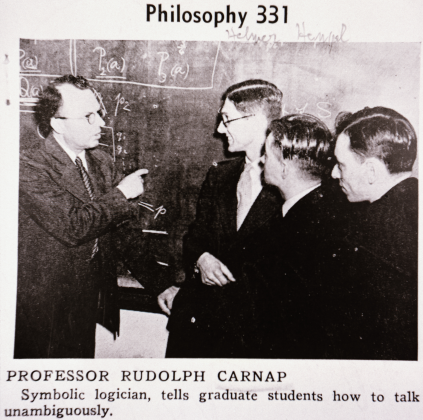
Ramified Type Theory (Russell)

Out of an abundance of caution, **Russell** eliminated value-ranges and *ramified* the hierarchy of propositional functions by their order.

- The terms of elementary propositions we will call *individuals*; these form the first or lowest type.
- Elementary propositions together with such as contain only individuals as apparent variables we will call *first-order propositions*. These form the second logical type.
- A function whose argument is an individual and whose value is ... a first-order proposition will be called a *first-order function*.
- A function involving a first-order function ... as apparent variable will be called a *second-order function*, and so on.

This resulted in a more complicated hierarchy of functions, determined also by the *expressions* used to define them.

Simple Type Theory



Simple Type Theory (Carnap, Gödel)

Carnap (1929) and **Gödel** (1931) returned to Frege's simple hierarchy of functions. As formulated by **Church** (1940):

- *types*: ι, o, $\alpha \to \beta$
- *terms*: $x : \alpha,\ b : \beta,\ \lambda x.b : \alpha \to \beta$
- *formulas* (terms of type o): $\neg\varphi,\ \varphi \Rightarrow \psi$, ..., $\exists_{x:\alpha}\varphi,\ \forall_{x\alpha}\varphi$
- *rules of deduction for formulas*: $\vartheta_1, \ldots, \vartheta_n \vdash \varphi$.

There is also the subsystem of λ-**calculus**, with only the types, terms, and equations between terms, such as

$$(\lambda x.f(x))(a) = f(a)\,.$$

Dependent Type Theory (Howard, Martin-Löf, Tait)

Dependent type theory replaces the formulas and deductions by further type and term constructors.

- *types*: X, 0, 1, $A + B$, $A \times B$, $A \to B$
- *terms*: x, $*$, $[a, b]$, $\langle a, b \rangle$, $\lambda x.b(x)$
- *dependent types*: $x : A \vdash B(x)$
- *sum types*: $\sum_{x:A} B(x)$
- *product types*: $\prod_{x:A} B(x)$

As a theory of *constructions* it is more expressive than λ-calculus.

It can even be seen as a system of *logic*.

Propositions as Types

There are, at first blush, two kinds of construction involved: constructions of proofs of some proposition and constructions of objects of some type. But I will argue that, from the point of view of foundations of mathematics, there is no difference between the two notions. A proposition may be regarded as a type of object, namely, the type of its proofs. Conversely, a type A may be regarded as a proposition, namely, the proposition whose proofs are the objects of type A. So a proposition A is true just in case there is an object of type A.

W.W. Tait

The Curry-Howard Correspondence

Under PAT the type constructors act as logical operations.

0	1	$A + B$	$A \times B$	$A \to B$	$\sum_{x:A} B(x)$	$\prod_{x:A} B(x)$
$\bot$	$\top$	$\alpha \vee \beta$	$\alpha \wedge \beta$	$\alpha \Rightarrow \beta$	$\exists_{x:\alpha}\beta(x)$	$\forall_{x:\alpha}\beta(x)$

This logic has a **constructive character**, which can be described proof theoretically:

- a proof $s : \Sigma_{x:A}B(x)$ provides $a : A$ and a proof $b : B(a)$,
- a proof $p : \Pi_{x:A}\Sigma_{y:B}R(x,y)$ provides a function $f : A \to B$ and a proof $q : \Pi_{x:A}R(x, fx)$.

Martin-Löf Type Theory

Identity Types (Martin-Löf)

Martin-Löf (1973) added an *identity type*, for terms $a, b : X$,

$$\mathsf{Id}_X(a, b)$$

Its rules preserved the constructive character of the system.
But they also introduced some **intensionality**:

- terms $a, b : X$ identified by $p : \mathsf{Id}_X(a, b)$ remain distinct,
- there may be different $p, q : \mathsf{Id}_X(a, b)$,
- is there always a term $\alpha : \mathsf{Id}_{\mathsf{Id}_X(a,b)}(p, q)$?

This system was used in computer proof systems like **Coq** because of its good computational properties, but its meaning remained somewhat mysterious ...

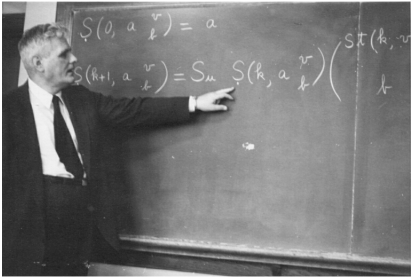
The Topological Interpretation: Simple Types (Scott)

Church had shown that a numerical function is *computable* iff it is *definable* in the simply-typed λ-calculus.

Scott showed how to interpret computability as **continuity**:

$$\text{types} \quad \rightsquigarrow \quad \text{spaces}$$

$$\text{terms} \quad \rightsquigarrow \quad \text{continuous functions}$$



The Homotopy Interpretation: Identity Types

Let us **extend the topological interpretation** to identity types!

$$\begin{array}{rcl} \text{types } X & \rightsquigarrow & \text{spaces} \\ \text{terms } t : X \to Y & \rightsquigarrow & \text{continuous functions} \\ \text{identities } p : \mathrm{Id}_X(a,b) & \rightsquigarrow & \text{paths } p : a \sim b \end{array}$$

In topology, a **path** $p : a \sim b$ from point a to point b in a space X is a continuous function

$$p : [0,1] \to X$$

with $p(0) = a$ and $p(1) = b$.

Homotopy

The relation $a \sim b$ satisfies the **laws of identity**,

$$a \sim a$$
$$a \sim b \;\Rightarrow\; b \sim a$$
$$a \sim b,\; b \sim c \;\Rightarrow\; a \sim c$$

Homotopy

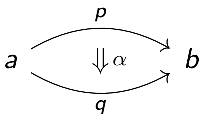
The relation $a \sim b$ satisfies the **laws of identity**,

$$r : a \sim a$$
$$p : a \sim b \;\Rightarrow\; p^{-1} : b \sim a$$
$$p : a \sim b,\; q : b \sim c \;\Rightarrow\; p.q : a \sim c$$

But the paths $p : a \sim b$ also satisfy **higher laws** like,

$$\alpha : p.(q.r) \approx (p.q).r$$

Such higher paths are called **homotopies**,



and they satisfy **even higher laws** ....

The Homotopy Interpretation: Identity Types

Identity types also endowed each type X with *higher structure*.

$$a, b : X$$
$$p, q : \mathsf{Id}_X(a, b)$$
$$\alpha, \beta : \mathsf{Id}_{\mathsf{Id}_X(a,b)}(p, q)$$
$$\ldots$$

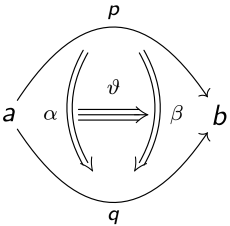
Under the *homotopy interpretation* these higher structures agree:

$$\begin{aligned}
X &\quad\rightsquigarrow\quad \text{space} \\
a, b : X &\quad\rightsquigarrow\quad \text{points of } X \\
p : \mathsf{Id}_X(a, b) &\quad\rightsquigarrow\quad \text{paths } p : a \sim b \\
\alpha : \mathsf{Id}_{\mathsf{Id}_X(a,b)}(p, q) &\quad\rightsquigarrow\quad \text{homotopies } \alpha : p \approx q \\
&\ldots
\end{aligned}$$

The Homotopy Interpretation: ∞-Groupoids

Theorem (Lumsdaine, van den Berg-Garner)

The identity types of a type X form an ∞-groupoid.

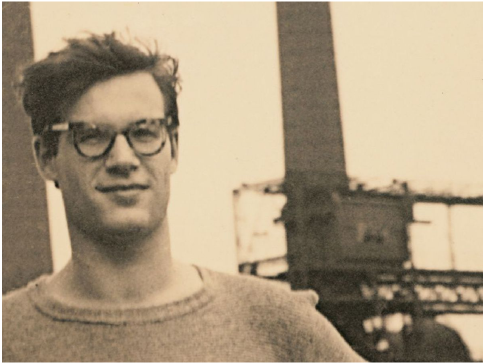


An ∞-groupoid is an ∞-category in which all arrows are isos.

The Homotopy Interpretation: ∞-groupoids

The points, paths, homotopies, ... in a space X were the original examples of ∞-groupoids, which first arose in Grothendieck's famous **homotopy hypothesis**:

Homotopy types of spaces are equivalent to ∞-groupoids

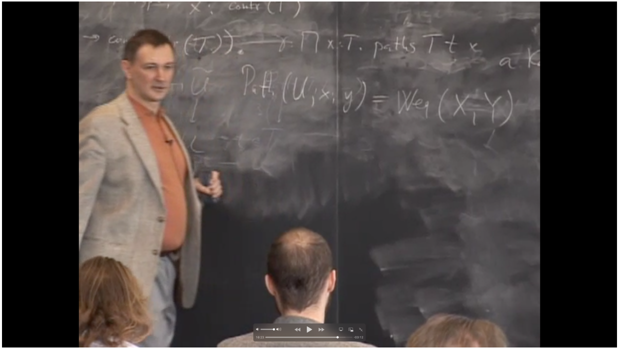


Univalent Foundations

Fields medalist Vladimir Voevodsky (IAS) had also arrived at related ideas while working on computer-checked proofs.

He proposed the *Univalence Axiom* in a lecture at CMU in 2010.

$$\mathsf{Id}(X, Y) \simeq (X \simeq Y)$$



Oberwolfach

A meeting was held at the Oberwolfach Mathematical Research Institute with Martin-Löf, Voevodsky, and others.

Oberwolfach

There the HoTT-Coq library was begun by Andrej Bauer.

Higher Inductive Types (HITs) were invented by Lumsdaine, Shulman, and Warren.

HITs can be used to represent spaces like:

- the **sphere** S^1 which parametrizes *loops* $\ell : x \sim x$ in a type X,

$$(S^1 \to X) \simeq \Sigma_{x:X} \mathrm{Id}(x, x)$$

- the **truncation** $||X||_0$, the set $X/_\sim$ of *connected components*.

Fundamental Groups

The **fundamental group** $\pi_1(X)$ of a space X was introduced by Henri Poincaré in 1895 in the influential paper *Analysis situs*. For $* \in X$ it consists of all loops $\ell : * \sim *$, up to homotopy.

Homotopy Groups of Spheres

Shulman calculated the fundamental group of the sphere S^1,

$$\pi_1(S^1) \;\simeq\; ||S^1 \dot{\to} S^1||_0 \;\simeq\; \mathsf{Id}_{S^1}(*, *) \;\simeq\; \mathbb{Z}\,,$$

and formalized the proof in HoTT-Coq.

This was the first of many benchmark calculations of *homotopy groups of spheres*.

The **higher spheres** S^n are also HITs. The higher **homotopy groups** $\pi_k(S^n)$ are then defined as the set of all *pointed maps* $S^k \dot{\to} S^n$, identified up to homotopy:

$$\pi_k(S^n) \;=\; ||S^k \dot{\to} S^n||_0$$

IAS Special Year on Univalent Foundations

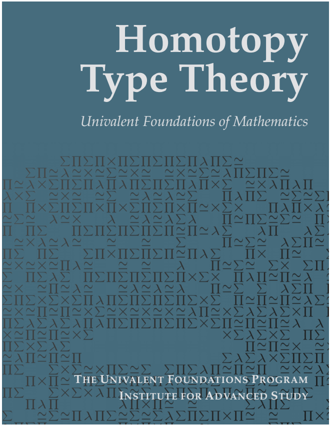
A special year on Univalent Foundations of Mathematics was held in 2012-13 at the Institute for Advanced Study, organized by Awodey, Coquand, and Voevodsky.

IAS Special Year on Univalent Foundations

IAS Special Year on Univalent Foundations

The HoTT Book

A book was jointly authored by the participants of the Special Year.



Thousands of copies have been sold (at cost), and many thousands more have been downloaded for free online.

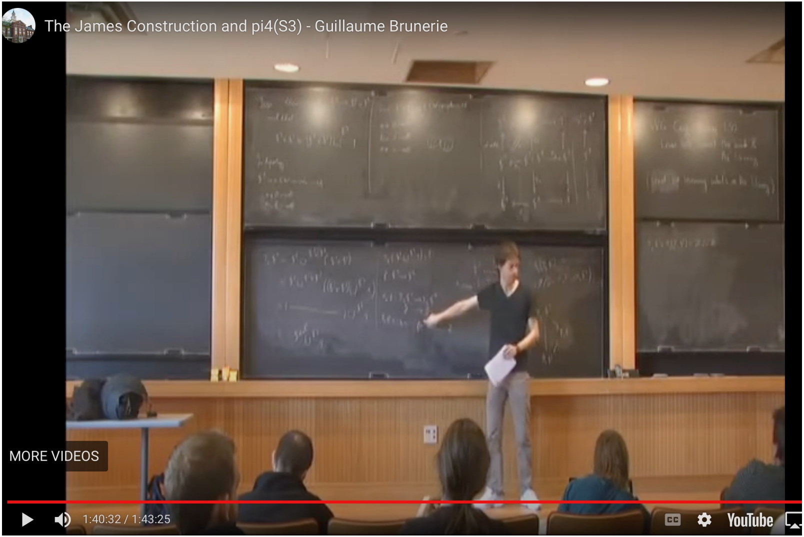
An Open Problem: Computation of $\pi_4(S^3)$

At the end of the Special Year, Brunerie calculated the 4th homotopy group of the 3-sphere in HoTT to be

$$\pi_4(S^3) \;\cong\; \mathbb{Z}/n\mathbb{Z}\,,$$

a classical result from homotopy theory.

Although the proof was constructive, the value of n could not be *computed* from the proof without a **constructive implementation** of univalence and HITs.


The James Construction and pi4(S3) - Guillaume Brunerie
MORE VIDEOS
1:40:32 / 1:43:25
YouTube

Brunerie's "Perfect World"

So what we get is that $\pi_4(\mathsf{S}^3)$ *. . . is equal to* $\mathbb{Z} \bmod n$ *for* **this** *n. And this is one very concrete and non-trivial example of why we may want to have canonicity, because this n is a closed term of type* $\mathbb{Z}$*, defined with a lot of univalence and higher inductive types. So in a perfect world, if you formalize that in a proof assistant with a computational interpretation of univalence . . . you can just ask "what is the value of n?" and you will get* 2*.*

Guillaume Brunerie, 23 May 2013, IAS

Computation of Brunerie's Number

Since 2013:

1. Constructivity of Univalence and HITs

   Coquand and collaborators developed a constructive version of HoTT with univalence and HITs (2014-15).

2. Implementation in a computational proof assistant
3. Computation of $\pi_4(S^3)$

Computation of Brunerie's Number

Since 2013:

1. Constructivity of Univalence and HITs (2014-15) ✓
2. Implementation in a computational proof assistant

   A new proof assistant that computes with Univalence and HITs was developed on that basis (2019).
3. Computation of $\pi_4(S^3)$

Computation of Brunerie's Number

Since 2013:

1. Constructivity of Univalence and HITs (2014–15) ✓
2. Implementation in a computational proof assistant (2019) ✓
3. Computation of $\pi_4(\mathsf{S}^3)$

   Brunerie's IAS proof that, for some $n : \mathbb{Z}$,

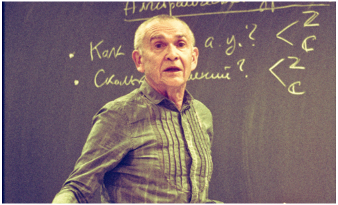
   $$\pi_4(\mathsf{S}^3) = \mathbb{Z}/_{n}\mathbb{Z}$$

   was formalized in the new proof assistant, and the value of $n = 2$ was computed from the proof (2022). ✓

Summary

1. Scott's insight that computability is modeled by continuity extends from the λ-calculus to constructive type theory.
2. Type theoretic constructions and judgements then become homotopy invariant structures and theorems.
3. Constructive proofs yield programs for calculating e.g. homotopy invariants in a computational proof system.
4. Classical foundations based on sets is a subsystem of this new constructive foundation based on homotopy types.

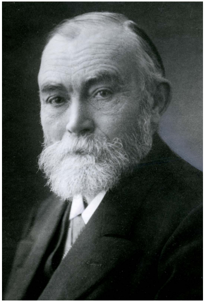
Yuri Manin



I am pretty strongly convinced that there is an ongoing reversal in the collective consciousness of mathematicians: the ... homotopical picture of the world becomes the basic intuition, and if you want to get a discrete set, then you pass to the set of connected components of a space defined only up to homotopy. That is, the Cantor points become continuous components ... almost from the start. Cantor's problems of the infinite recede to the background: from the very start, our images are so infinite that if you want to make something finite out of them, you must divide them by another infinity.

Interview with Mikhail Gelfand, 2008

Gottlob Frege



I am convinced that my Begriffsschrift will find successful application wherever particular value is placed on the rigor of proofs, as in the foundations of the differential and integral calculus. It seems to me that it would be even easier to extend the domain of this formal language to geometry. Only a few more symbols would need to be added for the intuitive relations occurring there. In this way, one would obtain a kind of analysis situs.

Preface to Begriffsschrift, 1879

Thanks!

For more information consult:

`HomotopyTypeTheory.org`

Some References

- A. Abel, A. Vezzosi and A. Mörtberg. Cubical Agda: A dependently typed programming language with univalence and higher inductive types. J. Functional Programming (2019).
- S. Awodey, T. Coquand. Univalent foundations and the large scale formalization of mathematics. The IAS Letter (Spring 2013).
- S. Awodey, M. Warren. Homotopy theoretic models of identity types, Mathematical Proceedings of the Cambridge Philosophical Society (2009).
- M. Bezem, T. Coquand, and S. Huber. A model of type theory in cubical sets. TYPES 2014.
- G. Brunerie. The James construction and $\pi_4(S^3)$. Institute for Advanced Study, March 2013.
- C. Cohen, T. Coquand, S. Huber and A. Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. TYPES 2015.
- B. van den Berg, R. Garner. Types are weak ω-groupoids, Proceedings of the London Mathematical Society (2011).
- A. Ljungström and A. Mörtberg. Formalizing $\pi_4(S^3) = \mathbb{Z}/_{2\mathbb{Z}}$ and Computing a Brunerie Number in Cubical Agda (2023).
- P. LeFanu Lumsdaine. Weak ω-categories from intensional type theory, Logical Methods in Computer Science (2010).
- The Univalent Foundations Program. Homotopy Type Theory: Univalent Foundations of Mathematics, Institute for Advanced Study (2013).

Appendix: The HoTT Interpretation of Type Dependency

A consequence of the interpretation of identity terms as *paths* is the interpretation of dependent types as *fibrations*.

A type family $x : X \vdash F(x)$ should be interpreted as a "continuously varying family of spaces", which we can take to be a *fiber bundle*, i.e. a continuous map:

$$x : X \vdash F(x) \quad \rightsquigarrow \quad \begin{array}{c} F \\ \big\downarrow \\ X \end{array}$$

Appendix: The HoTT Interpretation of Type Dependency

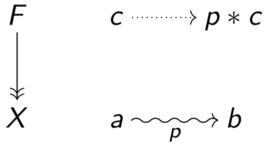
The rules for identity types permit the inference:

$$\frac{p : \mathsf{Id}_X(a,b) \qquad\qquad c : F(a)}{p * c : F(b)}$$

Logically, this just says the predicate $F(x)$ *respects identity*:

$$\mathsf{Id}_X(a,b)\ \&\ F(a)\ \Rightarrow\ F(b)$$

Topologically, it is the *path lifting property* of **fibrations**:



(To lift $p : a \sim b$ apply the lifting to the pathspace $F^I \twoheadrightarrow X$.)