Type Theory and Homotopy
I. Constructions and Dependence

Alex Kavvos

Panhellenic Logic Symposium, 6–10 July 2022

University of BRISTOL

I. Intuitionism and Constructions

Intuitionism, Constructivism, and Type Theory

- Many different philosophies: Brouwerian intuitionism, Heyting arithmetic, Russian constructivism, Bishop-style mathematics, etc. (see Stanford Encyclopedia of Philosophy entries)
- One common feature:

To prove that a mathematical object exists you must show how to construct it.

- In particular, the details of the construction matter.
- Modern algebra: the structure of an isomorphism matters.
- **Martin-Löf Type Theory** (MLTT) was created as a formalization of Bishop-style constructive mathematics.
- Less focus on **truth**, more focus on **proof**.
- The **law of the excluded middle** (LEM) $\phi \vee \neg\phi$ is rejected.

Constructions

Let $A, B, \ldots$ be sets.

$$\mathbf{0} \stackrel{\text{def}}{=} \emptyset \qquad \mathbf{1} \stackrel{\text{def}}{=} \{*\}$$

$$A \times B \stackrel{\text{def}}{=} \{(a, b) \mid a \in A \text{ and } b \in B\} \quad A \to B \stackrel{\text{def}}{=} \{f \mid f : A \to B\}$$

$$A + B \stackrel{\text{def}}{=} \{(1, a) \mid a \in A\} \cup \{(2, b) \mid b \in B\}$$

Let $\neg A \stackrel{\text{def}}{=} A \to \mathbf{0}$.

Example

- $(x, y) \mapsto x \in (A \times B) \to A$
- $x \mapsto (y \mapsto (x, y)) \in A \to (B \to A \times B)$
- $\lambda x.\, \lambda y.\, (x, y) \in A \to (B \to A \times B)$
- $\lambda(x, v).\, \begin{cases} (1, (x, b)) & \text{if } v = (1, b) \\ (2, (x, c)) & \text{if } v = (2, c) \end{cases} \in A \times (B + C) \to (A \times B) + (A \times C)$
- $\lambda a.\, \lambda f.\, f(a) \in A \to \neg\neg A$

Dependence

Let $(B_a)_{a \in A}$ be a **family** of sets.

$$(a : A) \times B_a \overset{\text{def}}{=} \sum_{a \in A} B_a \overset{\text{def}}{=} \{(a, b) \mid a \in A \text{ and } b \in B_a\}$$

$$(a : A) \to B_a \overset{\text{def}}{=} \prod_{a \in A} B_a \overset{\text{def}}{=} \left\{ f : A \to \bigcup_{a \in A} B_a \mid f(a) \in B_a \text{ for all } a \in A \right\}$$

Given a **constant** family of sets $(B)_{a \in A}$ we have

$$(a : A) \times B = A \times B \qquad\qquad (a : A) \to B = A \to B$$

Example

Let $P_n \overset{\text{def}}{=} \begin{cases} \{*\} & \text{if n is prime} \\ \emptyset & \text{otherwise} \end{cases}$

- $(11, *) \in (n : \mathbb{N}) \times P_n$, but $(4, *) \notin (n : \mathbb{N}) \times P_n$
- $\lambda n.\ \text{if } n \text{ is prime then } (1, *) \text{ else } (2, \mathrm{id}_\emptyset) \in (n : \mathbb{N}) \to P_n + \neg P_n$

II. Martin-Löf Type Theory

Martin-Löf Type Theory (MLTT)

- Invented by Per Martin-Löf in the late 1960s.
- A formal theory in **natural deduction** style.
- Every term in the theory needs to have a **type**.
- There are **no propositions**, only types.
  Every term is a **construction** which proves its **type**.

  types = predicates
  terms = proofs

- ZFC: engine (first-order logic) + fuel (axioms)
  MLTT: “engine and fuel all in one” (Pieter Hofstra, 1975–2022)

Judgements

Six distinct kinds of **judgement**:

Γ ctx	Γ is a context
$\Gamma \vdash A$ type	A is a type in context Γ
$\Gamma \vdash M : A$	M is a term of type A in context Γ
$\Gamma \equiv \Delta$ ctx	Γ and Δ are **definitionally** equal contexts
$\Gamma \vdash A \equiv B$ type	A and B are **definitionally** equal types
$\Gamma \vdash M \equiv N : A$	M and N are **definitionally** equal terms

The equality judgements have rules that make them

- equivalence relations, e.g. $\dfrac{\Gamma \vdash A \equiv B \text{ type}}{\Gamma \vdash B \equiv A \text{ type}}$
- congruences, e.g.

$$\frac{\Gamma \vdash A_1 \equiv A_2 \text{ type} \qquad \Gamma, x : A_1 \vdash B_1 \equiv B_2 \text{ type}}{\Gamma \vdash (x : A_1) \to B_1 \equiv (x : A_2) \to B_2 \text{ type}}$$

Contexts, variables, conversion

A **context** is a list of variables and their types.

$$\frac{}{\cdot \ \mathsf{ctx}} \qquad\qquad \frac{\Gamma\ \mathsf{ctx} \qquad \Gamma \vdash A\ \mathsf{type}}{\Gamma, x : A\ \mathsf{ctx}}$$

Variables stand for terms.
If I have a variable I can use it as a term:

$$\frac{\Gamma, x : A, \Delta\ \mathsf{ctx}}{\Gamma, x : A, \Delta \vdash x : A}$$

We can always replace definitionally equals by equals.
The **type conversion** rule:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A \equiv B\ \mathsf{type}}{\Gamma \vdash M : B}$$

What is a type?

It is a classifier of terms.

Terms of a certain type have an **interface**: a specification of how they can be created and consumed.

Ingredients of a type

- a **formation** rule (when can I form this type?)
- an **introduction** rule (how do I make terms of this type?)
- an **elimination** rule (how do I use terms of this type?)
- a **computation** rule (how do I calculate with its elements?)
- a **uniqueness** rule (what do terms of this type look like?)

Sometimes computation rules are called β **rules**
and uniqueness rules η **rules**.

Dependent function types / Π types

formation
$$\frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash (x : A) \to B \text{ type}}$$

introduction
$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.\, M : (x : A) \to B}$$

elimination
$$\frac{\Gamma \vdash M : (x : A) \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M(N) : B[N/x]}$$

computation
$$\frac{\Gamma, x : A \vdash M : B \qquad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x : A.\, M)(N) \equiv M[N/x] : B[N/x]}$$

uniqueness
$$\frac{\Gamma \vdash M : (x : A) \to B}{\Gamma \vdash M \equiv \lambda x : A.\, M(x) : (x : A) \to B}$$

Dependent sum types / Σ types

formation

$$\frac{\Gamma, x : A \vdash B \text{ type}}{\Gamma \vdash (x : A) \times B \text{ type}}$$

introduction

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash (M, N) : (x : A) \times B}$$

elimination

$$\frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash \mathrm{pr}_1(M) : A} \quad \frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash \mathrm{pr}_2(M) : B[\mathrm{pr}_1(M)/x]}$$

computation

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash \mathrm{pr}_1((M, N)) \equiv M : A}$$

$$\frac{\Gamma, x : A \vdash B \text{ type} \qquad \Gamma \vdash M : A \qquad \Gamma \vdash N : B[M/x]}{\Gamma \vdash \mathrm{pr}_2((M, N)) \equiv N : B[M/x]}$$

uniqueness

$$\frac{\Gamma \vdash M : (x : A) \times B}{\Gamma \vdash M \equiv (\mathrm{pr}_1(M), \mathrm{pr}_2(M)) : (x : A) \times B}$$

Coproducts (disjoint unions)

form.
$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A + B \text{ type}}$$

intro.
$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathsf{inl}(M) : A + B} \quad \frac{\Gamma \vdash N : B}{\Gamma \vdash \mathsf{inr}(N) : A + B}$$

elim.
$$\frac{\begin{array}{c}\Gamma \vdash M : A + B \qquad \Gamma, c : A + B \vdash C \text{ type} \\ \Gamma, x : A \vdash P : C[\mathsf{inl}(x)/c] \qquad \Gamma, y : B \vdash Q : C[\mathsf{inr}(y)/c]\end{array}}{\Gamma \vdash \mathsf{case}_{[c.C]}(M; x.\, P; y.\, Q) : C[M/c]}$$

comp.
$$\frac{\begin{array}{c}\Gamma \vdash M : A + B \\ \Gamma, c : A + B \vdash C \text{ type} \qquad \Gamma, x : A \vdash P : C[\mathsf{inl}(x)/c] \\ \Gamma, y : B \vdash Q : C[\mathsf{inr}(y)/c] \qquad \Gamma \vdash E : A\end{array}}{\Gamma \vdash \mathsf{case}_{[c.C]}(\mathsf{inl}(E); x.\, P; y.\, Q) \equiv P[E/x] : C[\mathsf{inl}(E)/c]}$$

Natural numbers

form.

$$\frac{}{\Gamma \vdash \mathsf{Nat}\ \mathsf{type}}$$

intro.

$$\frac{}{\Gamma \vdash \mathsf{zero} : \mathsf{Nat}} \quad \frac{\Gamma \vdash N : \mathsf{Nat}}{\Gamma \vdash \mathsf{succ}(N) : \mathsf{Nat}}$$

elim.

$$\frac{\begin{array}{cc} \Gamma \vdash N : \mathsf{Nat} & \Gamma, n : \mathsf{Nat} \vdash C\ \mathsf{type} \\ \Gamma \vdash P : C[\mathsf{zero}/n] & \Gamma, n : \mathsf{Nat}, c : C \vdash Q : C[\mathsf{succ}(n)/n] \end{array}}{\Gamma \vdash \mathsf{rec}_{[c.C]}(N; P; n, c.\, Q) : C[N/n]}$$

comp.

$$\frac{\ldots}{\Gamma \vdash \mathsf{rec}_{[c.C]}(\mathsf{zero}; P; n, c.\, Q) \equiv P : C[\mathsf{zero}/n]}$$

$$\frac{\ldots}{\Gamma \vdash \mathsf{rec}_{[c.C]}(\mathsf{succ}(x); P; n, c.\, Q) \equiv Q[x, \mathsf{rec}(x; P; n, c.\, Q)/n, c] : C[\mathsf{succ}(x)/n]}$$

Metatheory (I)

Let $\mathcal{J}$ stand for either A type or $M : A$.

Theorem (Weakening)

The following rule is admissible: $\dfrac{\Gamma, \Delta \vdash \mathcal{J} \qquad \Gamma \vdash A \text{ type}}{\Gamma, x : A, \Delta \vdash \mathcal{J}}$

Theorem (Substitution / Cut)

The following rule is admissible: $\dfrac{\Gamma \vdash M : A \qquad \Gamma, x : A, \Delta \vdash \mathcal{J}}{\Gamma, \Delta[M/x] \vdash \mathcal{J}[M/x]}$

Theorem

There is a set-theoretic model of MLTT with Π, Σ, Nat, *and* $+$ *types.*

The model can also be constructed in CZF (constructive ZF).

Corollary: the theory is consistent (if the ambient metatheory is).

Metatheory (II)

Theorem (Canonicity)

Let $\vdash M : C$. *Then:*

- *if* $C \equiv A + B$ *then either* $\vdash M \equiv \mathsf{inl}(P) : A + B$ *for some* $\vdash P : A$ *or* $\vdash N \equiv \mathsf{inr}(Q) : A + B$ *for some* $\vdash Q : B$,
- *if* $C \equiv \mathsf{Nat}$ *then* $\vdash M \equiv succ^n(\mathsf{zero}) : \mathsf{Nat}$ *for some* $n \in \mathbb{N}$
- *if* $C \equiv (x : A) \times B$ *then* $\vdash M \equiv (P, Q) : (x : A) \times B$ *for some* $\vdash P : A$ *and* $\vdash Q : B[P/x]$

Moreover, finding the "canonical form" of such terms is computable.

Theorem (Normalization)

Given Γ, M, N *and* A, *it is decidable whether* $\Gamma \vdash M \equiv N : A$.

Theorem (Decidability)

Given Γ, *and any judgement* $\mathcal{J}$, *it is decidable whether* $\Gamma \vdash \mathcal{J}$.

These properties give MLTT its computational flavour.

III. Examples

Propositional constructions

$\boxed{\text{Types are propositions. Terms are proofs.}}$

Define:

$$\wedge \overset{\text{def}}{=} \times \qquad\qquad \vee \overset{\text{def}}{=} +$$

Given $\vdash A, B$ type we have

- $\vdash \lambda x.\, \lambda y.\, x : A \to B \to A$
- $\vdash \lambda x.\, \lambda y.\, (x, y) : A \to B \to A \wedge B$
- $\vdash \lambda p.\, (\mathrm{pr}_2(p), \mathrm{pr}_1(p)) : A \wedge B \to B \wedge A$
- $\vdash \lambda u.\, \mathrm{case}(u; x.\, \mathrm{inr}(x); y.\, \mathrm{inl}(y)) : A \vee B \to B \vee A$

Theorem (Curry-Howard correspondence)

All intuitionistically valid formulas/types are inhabited.

Addition

Let $\Gamma \stackrel{\text{def}}{=} x : \mathsf{Nat}, y : \mathsf{Nat}$.

$$\dfrac{\dfrac{}{\Gamma \vdash x : \mathsf{Nat}} \qquad \dfrac{}{\Gamma \vdash y : \mathsf{Nat}} \qquad \dfrac{\dfrac{}{\Gamma, n : \mathsf{Nat}, c : \mathsf{Nat} \vdash n : \mathsf{Nat}}}{\Gamma, n : \mathsf{Nat}, c : \mathsf{Nat} \vdash \mathsf{succ}(n) : \mathsf{Nat}}}{\Gamma \vdash \mathsf{rec}_{[\_.\mathsf{Nat}]}(x; y; n, c.\, \mathsf{succ}(c)) : \mathsf{Nat}}$$

So we can define

$$\vdash \mathsf{add} = \lambda x.\, \lambda y.\, \mathsf{rec}(x; y; n, c.\, \mathsf{succ}(n)) : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$$

and compute

$$\begin{aligned} y : \mathsf{Nat} &\vdash \mathsf{add}(\mathsf{zero})(y) \equiv y : \mathsf{Nat} \\ y : \mathsf{Nat} &\vdash \mathsf{add}(\mathsf{succ}(\mathsf{zero}))(y) \equiv \mathsf{succ}(y) : \mathsf{Nat} \end{aligned}$$

and so on.

A familiar construction (I)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\dfrac{\dfrac{\overline{x : A, y : B \vdash R(x, y)\ \text{type}}}{x : A \vdash (y : B) \times R(x, y)\ \text{type}}}{\vdash (x : A) \to (y : B) \times R(x, y)\ \text{type}}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.

Similarly, recalling that $A \to B \overset{\text{def}}{=} (x : A) \to B$, we have

$$\dfrac{\dfrac{\vdots}{f : A \to B, x : A \vdash R(x, f(x))\ \text{type}}}{\vdash (f : A \to B) \times ((x : A) \to R(x, f(x)))\ \text{type}}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

A familiar construction (II)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\vdash (x : A) \to (y : B) \times R(x, y) \text{ type}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.
Similarly, recalling that $A \to B \stackrel{\text{def}}{=} (x : A) \to B$, we have

$$\vdash (f : A \to B) \times ((x : A) \to R(x, f(x))) \text{ type}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

$$\begin{aligned}\Gamma \vdash {?} : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))\end{aligned}$$

A familiar construction (II)

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\vdash (x : A) \to (y : B) \times R(x, y) \text{ type}$$

This is essentially $\forall x : A.\ \exists y : B.\ R(x, y)$.
Similarly, recalling that $A \to B \overset{\text{def}}{=} (x : A) \to B$, we have

$$\vdash (f : A \to B) \times ((x : A) \to R(x, f(x))) \text{ type}$$

This is essentially $\exists f : A \to B.\ \forall x : A.\ R(x, f(x))$.

$$\begin{aligned}\Gamma \vdash ? : ((x : A) \to (y : B) \times R(x, y))& \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))&\end{aligned}$$

Indeed, this is the **type-theoretic "axiom" of choice**:

$$\begin{aligned}\Gamma \vdash \lambda g.\,(\lambda x.\,\mathrm{pr}_1(g(x)), \lambda x.\,\mathrm{pr}_2(g(x))) : ((x : A) \to (y : B) \times R(x, y))& \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))&\end{aligned}$$

The type-theoretic "axiom" of choice

Let $\Gamma \vdash A, B$ type, and $x : A, y : B \vdash R(x, y)$ type. Then

$$\Gamma \vdash \lambda g.\, (\lambda x.\, \mathrm{pr}_1(g(x)), \lambda x.\, \mathrm{pr}_2(g(x))) : ((x : A) \to (y : B) \times R(x, y)) \\ \to ((f : A \to B) \times ((x : A) \to R(x, f(x))))$$

Suppose $g : (x : A) \to (y : B) \times R(x, y)$. Then clearly

$$f_g \stackrel{\text{def}}{=} \lambda x : A.\, \underbrace{\mathrm{pr}_1(\overbrace{g(x)}^{(y:B)\times R(x,y)})}_{B} : A \to B$$

$$h_g \stackrel{\text{def}}{=} \lambda x : A.\, \underbrace{\mathrm{pr}_2(\overbrace{g(x)}^{(y:B)\times R(x,y)})}_{R(x,\mathrm{pr}_1(g(x)))} : (x : A) \to R(x, \mathrm{pr}_1(g(x)))$$

But $f(x) \equiv \mathrm{pr}_1(g(x))$, so this type is equal to $(x : A) \to R(x, f(x))$.
Hence $\lambda g.\, (f_g, h_g)$ has the right type.

Summary

- MLTT is a formal theory of **constructions** and **dependence**.
- It has very good metatheoretic and computational properties.
- It is inherently “constructive” (for some sense of the word).

Tomorrow: equality as a proposition/type.

References

- Martin Hofmann, *Syntax and Semantics of Dependent Types*, Semantics and Logics of Computation (Andrew M. Pitts and P. Dybjer, eds.), Cambridge University Press, 1997, pp. 79–130.
- Per Martin-Löf, *An Intuitionistic Theory of Types: Predicative Part*, Logic Colloquium '73 (H. E. Rose and J. C. Shepherdson, eds.), Studies in Logic and the Foundations of Mathematics, no. 80, Elsevier, Bristol, 1975, pp. 73–118.
- ______, *Constructive mathematics and computer programming*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences **312** (1984), no. 1522, 501–518.
- Bengt Nordström, Kent Petersson, and Jan M. Smith, *Programming in Martin-Löf's Type Theory: an Introduction*, Oxford University Press, 1990.

Type Theory and Homotopy
II. Identity

Alex Kavvos

Panhellenic Logic Symposium, 6–10 July 2022

University of BRISTOL

Equality

Recall that we could define

$$\vdash \mathsf{add} = \lambda x.\, \lambda y.\, \mathsf{rec}(x; y; n, c.\, \mathsf{succ}(n)) : \mathsf{Nat} \to \mathsf{Nat} \to \mathsf{Nat}$$

and compute that

$$y : \mathsf{Nat} \vdash \mathsf{add}(\mathsf{zero})(y) \equiv y : \mathsf{Nat}$$

It is **not** the case that

$$x : \mathsf{Nat} \vdash \mathsf{add}(x)(\mathsf{zero}) \equiv x : \mathsf{Nat}$$

$\equiv$ only allows unfolding of definitions, **not** non-trivial theorems.

For that we need to introduce the **identity type**.

I. Identity Types

Intensional Identity Types

form. $$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash N : A}{\Gamma \vdash \mathrm{Id}_A(M, N) \text{ type}}$$

intro. $$\frac{\Gamma \vdash M : A}{\Gamma \vdash \mathrm{refl}(M) : \mathrm{Id}_A(M, M)}$$

elim. $$\frac{\begin{array}{c}\Gamma, x : A, y : A, p : \mathrm{Id}_A(x, y) \vdash B \text{ type} \\ \Gamma \vdash P : \mathrm{Id}_A(M, N) \qquad \Gamma, z : A \vdash Q : B[z, z, \mathrm{refl}(z)/x, y, p]\end{array}}{\Gamma \vdash \mathrm{J}_{[x,y,p.B]}(P; z.\, Q) : B[M, N, P/x, y, p]}$$

comp. $$\frac{\begin{array}{c}\Gamma, x : A, y : A, p : \mathrm{Id}_A(x, y) \vdash B \text{ type} \\ \Gamma, z : A \vdash Q : B[z, z, \mathrm{refl}(z)/x, y, p]\end{array}}{\Gamma \vdash \mathrm{J}(\mathrm{refl}(M); z.\, Q) \equiv Q[M/z] : B[M, M, \mathrm{refl}(M)/x, y, p]}$$

Because of the **type conversion** and **congruence** rules we always have

$$\frac{\Gamma \vdash M \equiv N : A}{\Gamma \vdash \mathrm{refl}(M) : \mathrm{Id}_A(M, N)}$$

Some examples (I)

- Let $\vdash A$ type and $x : A \vdash P(x)$ type. We have:

  $$x, y : A, p : \mathrm{Id}_A(x, y) \vdash \mathsf{transp}(p) \equiv \mathsf{J}(p; z.\, \lambda w.\, w) : B(x) \to B(y)$$

  Informally:

  Let $x, y : A$ and $p : \mathrm{Id}_A(x, y)$. We want to construct a term of type $B(x) \to B(y)$. By elimination we may assume that $x \equiv y$, so it suffices to give a term $B(x) \to B(x)$. Take the identity function.

- Let $x : A \vdash f(x) : B$. Then $x, y : A \vdash \mathrm{Id}_B(f(x), f(y))$ type. We have

  $$x, y : A, p : \mathrm{Id}_A(x, y) \vdash \mathsf{ap}_f(p) \equiv \mathsf{J}(p; x.\, \mathsf{refl}(f(x))) : \mathrm{Id}_B(f(x), f(y))$$

  Informally:

  Let $x, y : A$ and $p : \mathrm{Id}_A(x, y)$. We want to show $\mathrm{Id}_B(f(x), f(y))$. By elimination we may assume that $x \equiv y$, so it suffices to construct a term of type $\mathrm{Id}_B(f(x), f(x))$. Take $\mathsf{refl}(f(x))$.

Some examples (II)

Here is an informal proof that there is a term of type

$$x : \mathsf{Nat} \vdash \mathsf{Id}_{\mathsf{Nat}}(\mathsf{add}(x)(\mathsf{zero}), x)\ \mathsf{type}$$

We proceed by induction on $x : \mathsf{Nat}$.

- If $x \equiv \mathsf{zero} : \mathsf{Nat}$, then $\mathsf{add}(x)(\mathsf{zero}) \equiv \mathsf{add}(\mathsf{zero})(\mathsf{zero}) \equiv \mathsf{zero}$. Hence it suffices to construct $\mathsf{refl}(\mathsf{zero}) : \mathsf{Id}_{\mathsf{Nat}}(\mathsf{zero}, \mathsf{zero})$.
- If $x \equiv \mathsf{succ}(y) : \mathsf{Nat}$ for some $y : \mathsf{Nat}$, then

  $$\mathsf{add}(x)(\mathsf{zero}) \equiv \mathsf{add}(\mathsf{succ}(y))(\mathsf{zero}) \equiv \mathsf{succ}(\mathsf{add}(y)(\mathsf{zero}))$$

  By the IH we have $p : \mathsf{Id}_{\mathsf{Nat}}(\mathsf{add}(y)(\mathsf{zero}), y)$. Hence

  $$\mathsf{ap}_{\mathsf{succ}(-)}(p) : \mathsf{Id}_{\mathsf{Nat}}(\underbrace{\mathsf{succ}(\mathsf{add}(y)(\mathsf{zero}))}_{\equiv\, \mathsf{add}(x)(\mathsf{zero})}, \underbrace{\mathsf{succ}(y)}_{\equiv\, x})$$

  So we have shown the inductive step.

Metatheory

Theorem

The following rule is admissible.

$$\frac{\vdash P : \mathrm{Id}_A(M, N)}{\vdash M \equiv N : A}$$

Any two propositionally equal terms in an **empty context** are also definitionally equal. (Hence the name 'intensional.')

This did not apply to our previous proof because $x : \mathrm{Nat}$ was free.

Theorem

There is a set-theoretic model of MLTT with Π, Σ, *Id,* Nat, *and* $+$ *types.*

Extensional Identity Types

One might argue that $x : \mathsf{Nat} \vdash \ldots : \mathsf{Id}_{\mathsf{Nat}}(\mathsf{add}(x)(\mathsf{zero}), x)$ should be promoted to a definitional equality

$$x : \mathsf{Nat} \vdash \mathsf{add}(x)(\mathsf{zero}) \equiv x : \mathsf{Nat}$$

Add **equality reflection** rule:

$$\frac{\Gamma \vdash P : \mathsf{Id}_A(M, N)}{\Gamma \vdash M \equiv N : A}$$

We then say we have **extensional identity types**. But then

- normalization is no longer decidable, and hence
- **type checking is no longer decidable**

So we are stuck with the 'bureaucracy' of intensional identity types.

But this is a fine type theory for computing by hand.

II. Homotopy

Identity types are very mysterious

Let $\vdash M, N : A$. Construct $\vdash \mathrm{Id}_A(M, N)$ type.

Now suppose $\vdash P, Q : \mathrm{Id}_A(M, N)$.

What is the meaning of the following type?

$$\vdash \mathrm{Id}_{\mathrm{Id}_A(M,N)}(P, Q) \text{ type}$$

Should the following **Uniqueness of Identity Proofs (UIP)** principle be inhabited for any type $\Gamma \vdash A$ type?

$$\vdash (x, y : A) \to (p, q : \mathrm{Id}_A(x, y)) \to \mathrm{Id}_{\mathrm{Id}_A(x,y)}(p, q) \text{ type} \tag{UIP}$$

It's certainly true in the set-theoretic model!

Theorem (Hofmann-Streicher, 1998)

There is a model of MLTT in which the above principle of ***uniqueness of identity proofs*** *(UIP) is* ***not*** *true.*

Groupoids

Definition

A **groupoid** $\mathcal{G}$ consists of

- a set of **objects** $\mathrm{ob}(\mathcal{G})$
- for $x, y \in \mathrm{ob}(\mathcal{G})$ a set of **isomorphisms** $\mathrm{Hom}(x, y)$
  We write $f : x \xrightarrow{\cong} y$ if $f \in \mathrm{Hom}(x, y)$.
- for each $x \in \mathrm{ob}(\mathcal{G})$ an **identity** $1_x \in \mathrm{Hom}(x, x)$
- for isos $f : x \xrightarrow{\cong} y$ and $g : y \xrightarrow{\cong} z$ a **composite**

$$g \circ f : x \xrightarrow{\cong} z$$

- for each iso $f : x \xrightarrow{\cong} y$ and **inverse iso** $f^{-1} : y \xrightarrow{\cong} x$, such that

$$f^{-1} \circ f = 1_x : x \xrightarrow{\cong} x \qquad f \circ f^{-1} = 1_y : y \xrightarrow{\cong} y$$

A **one-object** groupoid is ...

Groupoids

Definition

A **groupoid** $\mathcal{G}$ consists of

- a set of **objects** $\mathrm{ob}(\mathcal{G})$
- for $x, y \in \mathrm{ob}(\mathcal{G})$ a set of **isomorphisms** $\mathrm{Hom}(x, y)$
  We write $f : x \xrightarrow{\cong} y$ if $f \in \mathrm{Hom}(x, y)$.
- for each $x \in \mathrm{ob}(\mathcal{G})$ an **identity** $1_x \in \mathrm{Hom}(x, x)$
- for isos $f : x \xrightarrow{\cong} y$ and $g : y \xrightarrow{\cong} z$ a **composite**

$$g \circ f : x \xrightarrow{\cong} z$$

- for each iso $f : x \xrightarrow{\cong} y$ and **inverse iso** $f^{-1} : y \xrightarrow{\cong} x$, such that

$$f^{-1} \circ f = 1_x : x \xrightarrow{\cong} x \qquad f \circ f^{-1} = 1_y : y \xrightarrow{\cong} y$$

A **one-object** groupoid is …a group!

If $|\mathrm{Hom}(x, y)| \leq 1$ a groupoid is …

Groupoids

Definition

A **groupoid** $\mathcal{G}$ consists of

- a set of **objects** $\mathrm{ob}(\mathcal{G})$
- for $x, y \in \mathrm{ob}(\mathcal{G})$ a set of **isomorphisms** $\mathrm{Hom}(x, y)$
  We write $f : x \xrightarrow{\cong} y$ if $f \in \mathrm{Hom}(x, y)$.
- for each $x \in \mathrm{ob}(\mathcal{G})$ an **identity** $1_x \in \mathrm{Hom}(x, x)$
- for isos $f : x \xrightarrow{\cong} y$ and $g : y \xrightarrow{\cong} z$ a **composite**

$$g \circ f : x \xrightarrow{\cong} z$$

- for each iso $f : x \xrightarrow{\cong} y$ and **inverse iso** $f^{-1} : y \xrightarrow{\cong} x$, such that

$$f^{-1} \circ f = 1_x : x \xrightarrow{\cong} x \qquad f \circ f^{-1} = 1_y : y \xrightarrow{\cong} y$$

A **one-object** groupoid is …a group!

If $|\mathrm{Hom}(x, y)| \leq 1$ a groupoid is …an equivalence relation!

The Hofmann-Streicher groupoid model of type theory

Hofmann and Streicher interpreted MLTT as follows:

- $\vdash A$ type is interpreted by a groupoid $[\![A]\!]$.
- A type family/dependent type $x : A \vdash B$ type is interpreted by a **fibration** $[\![B]\!] : [\![A]\!] \to \mathbf{GPD}$ of groupoids.
- A term of type $x : A \vdash B$ type is a **section** of the fibration $[\![B]\!]$.
- The identity type $\vdash \mathrm{Id}_A(M, N)$ type is interpreted by the set of isomorphisms of the groupoid $[\![A]\!]$, i.e.

$$\mathrm{Hom}_{[\![A]\!]}([\![M]\!], [\![N]\!])$$

In this model there are types with **non-trivial identity types**.

But where do groupoids come from?

Paths

Let X be a (topological) space.

Definition

A **path** in space X is a continuous function $p : [0,1] \to X$.

Write $p : x \rightsquigarrow y$ if $p(0) = x$ and $p(1) = y$.

Given $p : x \rightsquigarrow y$ let $p^{-1} : y \rightsquigarrow x$ by $p^{-1}(t) \stackrel{\text{def}}{=} p(1-t)$.

Given $p : x \rightsquigarrow y$ and $q : y \rightsquigarrow z$ let

$$(p \cdot q)(t) \stackrel{\text{def}}{=} \begin{cases} p(2t) & \text{if } 0 \leq t \leq 1/2 \\ q(2t-1) & \text{if } 1/2 \leq t \leq 1 \end{cases}$$

Question: given

$$p : x \rightsquigarrow y \qquad q : y \rightsquigarrow z \qquad r : z \rightsquigarrow w$$

is the following true?

$$(p \cdot q) \cdot r \stackrel{?}{=} p \cdot (q \cdot r)$$

Homotopy

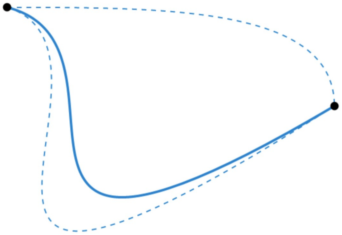
Let $f, g : X \to Y$ be continuous functions.

Definition

A **homotopy** H from f to g is a continuous function

$$H : X \times [0, 1] \to Y$$

such that $H(-, 0) = f$ and $H(-, 1) = g$.



Write $f \sim g$ if there is a homotopy from f to g.
$\sim$ is an equivalence relation.

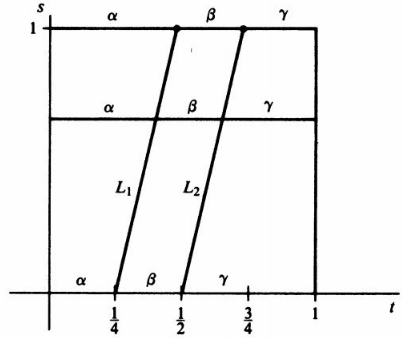
Associativity and Homotopy

Given

$$p : x \rightsquigarrow y \qquad q : y \rightsquigarrow z \qquad r : z \rightsquigarrow w$$

we have that

$$(p \centerdot q) \centerdot r \sim p \centerdot (q \centerdot r) : x \rightsquigarrow w$$



If $1_x : x \rightsquigarrow x$ and $1_y : y \rightsquigarrow y$ are constant paths then $p \centerdot 1_y \sim p \sim 1_x \centerdot p$.

The Fundamental Groupoid

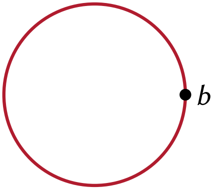
Let X be a space. Its **fundamental groupoid** $\pi(X)$ consists of

objects — the points of X

isomorphisms — equiv. classes $[p]$ of paths $p : x \rightsquigarrow y$ up to $\sim$

Taking only equivalence classes of **loops** $p : x \rightsquigarrow x$ at $x \in X$ gives the **fundamental group** $\pi(X, x)$ of X at x.

These are essential **algebraic invariants** of the space X.

$\mathbb{S}^1 \stackrel{\text{def}}{=}$ 

Theorem

$\pi(\mathbb{S}^1, b) \cong \mathbb{Z}$

∞-Groupoids

The fundamental insight:

$\boxed{\text{Why quotient at all?}}$

Definition

A **groupoid** $\mathcal{G}$ consists of

- a set of **objects** $\mathrm{ob}(\mathcal{G})$
- for $x, y \in \mathrm{ob}(\mathcal{G})$ a set of **isomorphisms** $\mathrm{Hom}(x, y)$

$\vdots$

Definition (sort of)

An ∞-**groupoid** $\mathcal{G}$ consists of

- a set of **0-cells** $\mathrm{ob}(\mathcal{G})$
- for $x, y \in \mathrm{ob}(\mathcal{G})$ an ∞-groupoid of **1-cells** $\mathrm{Hom}(x, y)$

$\vdots$

The Fundamental ∞-Groupoid

Let X be a space. Its **fundamental ∞-groupoid** $\pi_\infty(X)$ consists of

0-cells) the points of X

1-cells) paths $p : x \rightsquigarrow y$ between points

2-cells homotopies $H : p \sim q$ between paths

$\vdots$

Exact definition(s) tiresome to describe **analytically**.

Grothendieck's (1928–2014) dream, aka the **homotopy hypothesis**:

$\boxed{\infty\text{-groupoids} = \text{topological spaces up to homotopy}}$

Identity Types and Homotopy

The intended pun:

$$\boxed{\begin{array}{c}\text{types} = \text{spaces} = \infty\text{-groupoids}\\ \text{elements of the identity type} = \text{paths in the space}\end{array}}$$

For example, given $\vdash A$ type we can write down a term

$$\_ \centerdot \_ : (x, y, z : A) \to \mathrm{Id}_A(x, y) \to \mathrm{Id}_A(y, z) \to \mathrm{Id}_A(x, z)$$

Informal proof: Suppose $x, y, z : A$, $p : \mathrm{Id}_A(x, y)$, and $q : \mathrm{Id}_A(y, z)$. By the elimination rule we may assume that $x \equiv y$ and $y \equiv z$, so it suffices to define a term of type $\mathrm{Id}_A(x, x)$. Take $\mathrm{refl}(x)$.

Remember that because of the **computation rule** we have

$$\mathrm{refl}(x) \centerdot \mathrm{refl}(x) \equiv \mathrm{refl}(x)$$

Associativity of path composition

Given $x, y, z : A$ we can then define a term

$$\mathrm{assoc}_{xyz} : (p : \mathrm{Id}_A(x, y)) \to (q : \mathrm{Id}_A(y, z)) \to (r : \mathrm{Id}_A(z, w)) \to \mathrm{Id}_{\mathrm{Id}_A(x,w)}((p \cdot q) \cdot r, p \cdot (q \cdot r))$$

Informal proof. Given p, q, r as above we may assume that $x \equiv y \equiv z \equiv w$ and $p \equiv q \equiv r \equiv \mathrm{refl}(x)$. Thus, we only need a term of type

$$\mathrm{Id}_{\mathrm{Id}_A(x,x)}(\underbrace{(p \cdot q) \cdot r}_{\equiv\ \mathrm{refl}(x)}, \underbrace{p \cdot (q \cdot r)}_{\equiv\ \mathrm{refl}(x)})$$

and for that we may take $\mathrm{refl}(\mathrm{refl}(x))$.

Associativity of path composition

Given $x, y, z : A$ we can then define a term

$$\mathsf{assoc}_{xyz} : (p : \mathsf{Id}_A(x, y)) \to (q : \mathsf{Id}_A(y, z)) \to (r : \mathsf{Id}_A(z, w)) \to \mathsf{Id}_{\mathsf{Id}_A(x,w)}((p \cdot q) \cdot r, p \cdot (q \cdot r))$$

Informal proof. Given p, q, r as above we may assume that $x \equiv y \equiv z \equiv w$ and $p \equiv q \equiv r \equiv \mathsf{refl}(x)$. Thus, we only need a term of type

$$\mathsf{Id}_{\mathsf{Id}_A(x,x)}(\underbrace{(p \cdot q) \cdot r}_{\equiv\ \mathsf{refl}(x)}, \underbrace{p \cdot (q \cdot r)}_{\equiv\ \mathsf{refl}(x)})$$

and for that we may take $\mathsf{refl}(\mathsf{refl}(x))$.

This can be taken to its logical conclusion—see HoTT book:

> The elimination rule of the identity type generates the structure of an ∞-groupoid.

In other words, MLTT is a **synthetic** theory of ∞-groupoids.

Summary

- **Intensional identity types** allow proofs of non-trivial, non-definitional equalities in MLTT.
- Iterated identity types generate the structure of an ∞-groupoid.
- That is why sometimes the elimination rule for the identity type is known as **path induction**.
- MLTT can be seen as a synthetic theory of ∞-groupoids.

Tomorrow: homotopy levels; equivalence; higher inductive types.

References

Steve Awodey, *Type Theory and Homotopy*, Epistemology versus Ontology (P. Dybjer, Sten Lindström, Erik Palmgren, and G. Sundholm, eds.), Springer Netherlands, 2012, pp. 183–201.

Peter May, *A concise course in algebraic topology*, Chicago Lectures in Mathematics, University of Chicago Press, 1999.

Michael Shulman, *Homotopy Type Theory: A Synthetic Approach to Higher Equalities*, Categories for the Working Philosopher (Elaine Landry, ed.), Oxford University Press, 2017.

The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, 2013.

Type Theory and Homotopy
III. Equivalences, Univalence, and Quotients

Alex Kavvos

Panhellenic Logic Symposium, 6–10 July 2022

University of BRISTOL

Previously

Intensional identity types seem to support the following view:

- Types are **spaces** (up to homotopy).
- Terms are **points**.
- Elements of the identity type are **paths**.
- Everything given in a **synthetic** manner, not analytic.

This discovery is independently due to

- Awodey and Warren [Math. Proc. Camb. Philos. Soc. 2009]
- Vladimir Voevodsky (1966–2017) [Stanford lecture 2006]

Interpretation of TT into **simplicial sets**: Kapulkin and Lumsdaine, with thanks to Voevodsky [J. Eur. Math. Soc. 2018].

I. Homotopical structure of types

Homotopy Levels

Types are spaces; they have **higher-dimensional structure**.

Yet, some types do not. Let A be a type.

$$\begin{array}{rl} \textit{contractible} & \mathsf{isContr}(A) \stackrel{\text{def}}{=} (c : A) \times ((x : A) \to \mathsf{Id}_A(c, x)) \\ \text{proposition} & \mathsf{isProp}(A) \stackrel{\text{def}}{=} (x, y : A) \to \mathsf{Id}_A(x, y) \\ \text{set} & \mathsf{isSet}(A) \stackrel{\text{def}}{=} (x, y : A) \to (p, q : \mathsf{Id}_A(x, y)) \to \mathsf{Id}(p, q) \\ & \vdots \end{array}$$

In general, we define

$$\begin{aligned} \text{is-(-2)-type}(A) &\stackrel{\text{def}}{=} \mathsf{isContr}(A) \\ \text{is-(n+1)-type}(A) &\stackrel{\text{def}}{=} (x, y : A) \to \mathsf{isContr}(\mathsf{Id}_A(x, y)) \end{aligned}$$

Then

$$\text{is-(-1)-type}(A) \simeq \mathsf{isProp}(A) \qquad \text{is-0-type}(A) \simeq \mathsf{isSet}(A)$$

Propositions and Sets

Here is an unusual result:

Theorem (Hedberg, J. Func. Prog 1998)

*Let A be a type. If **identity is decidable**, i.e. if we have*

$$d : (x, y : A) \to \mathsf{Id}_A(x, y) + \neg\mathsf{Id}_A(x, y)$$

*then A **is a set**, i.e. we have a proof of isSet(A).*

Corollary

Nat *is a set.*

In some sense, all the maths we have done so far is **0-dimensional**!

II. Universes

Identity types are not good enough

Theorem (Jan Smith, J. Symb. Log. 1988)

The type correspondong to Peano's fourth axiom, i.e.

$$n : \mathsf{Nat} \vdash \mathsf{Id}_{\mathsf{Nat}}(0, \mathsf{succ}(n)) \to \mathbf{0} \text{ type}$$

*is **not** inhabited in MLTT with $\to$, $\times$, and identity types.*

Proof: construct a model of MLTT where types are subsingleton sets.

To prove Peano 4, we intuitively want to

1. construct a type family $n : \mathsf{Nat} \vdash B(n)$ type where

$$\begin{array}{rl} \vdash B(\mathsf{zero}) \text{ type} & \text{is inhabited} \\ n : \mathsf{Nat} \vdash B(\mathsf{succ}(n)) \text{ type} & \text{is empty, i.e. } \equiv \mathbf{0} \end{array}$$

2. assuming $n : \mathsf{Nat} \vdash P : \mathsf{Id}_{\mathsf{Nat}}(0, \mathsf{succ}(n))$ and $\vdash M : B(\mathsf{zero})$, obtain $n : \mathsf{Nat} \vdash \mathsf{transp}(P)(M) : B(\mathsf{succ}(n)) \equiv \mathbf{0}$

We cannot perform Step 1 because types are not terms.

Universes à la Russell

We introduce the **universe**, a **type of all (small) types**.

$$\frac{}{\Gamma \vdash \mathsf{U}\ \mathsf{type}} \qquad\qquad \frac{\Gamma \vdash A : \mathsf{U}}{\Gamma \vdash A\ \mathsf{type}}$$

plus one rule for each type constructor, e.g.

$$\frac{\Gamma \vdash A : \mathsf{U} \qquad \Gamma, x : A \vdash B : \mathsf{U}}{\Gamma \vdash (x : A) \to B : \mathsf{U}}$$

Caution. We must **avoid** the following to avoid paradoxes:

$$\frac{}{\Gamma \vdash \mathsf{U} : \mathsf{U}}$$

Types may then be constructed as terms of U (e.g. by induction).

If $A : \mathsf{U}$ then we say that A is a **small** type.

Homotopy equivalence

Definition

Two topological spaces X and Y are **homotopy-equivalent** if there are continuous functions $f : X \to Y$ and $g : Y \to X$ such that

$$g \circ f \sim 1_X \qquad\qquad f \circ g \sim 1_Y$$

where 1_X and 1_Y are the identity functions on X and Y.

We can model this synthetically in MLTT.

Type-theoretic Equivalences

Definition (Voevodsky)

We say that $f : A \to B$ is an **equivalence** just if

$$\mathsf{isEquiv}(f) \stackrel{\text{def}}{=} (y : B) \to \mathsf{isContr}((x : A) \times \mathsf{Id}_B(f(x), y))$$

This is a homotopically well-behaved notion of **isomorphism**.

For $A, B : \mathsf{U}$ define the type of **(type-theoretic) equivalences**

$$A \simeq B \stackrel{\text{def}}{=} (f : A \to B) \times \mathsf{isEquiv}(f)$$

We can use equivalences to **decompose** identity types.

E.g. for any $A, B : \mathsf{U}$ and $p, q : A \times B$:

$$\mathsf{Id}_{A \times B}(p, q) \simeq \mathsf{Id}_A(\mathsf{pr}_1(p), \mathsf{pr}_1(q)) \times \mathsf{Id}_B(\mathsf{pr}_2(p), \mathsf{pr}_2(q))$$

This can be done for most type formers of MLTT.

Univalence

Question:

$$\boxed{\text{What is an identity between types?}}$$

Voevodsky proposed adding the **univalence axiom** to MLTT:

$$\mathsf{ua} : (A, B : \mathsf{U}) \to (A \simeq B) \simeq \mathsf{Id}_{\mathsf{U}}(A, B)$$

This spoils the computational character of MLTT, but is a revolution:

$$\boxed{\text{isomorphic/equivalent types are identical}}$$

This principle is often used informally in maths ('abuse of notation'). E.g. the Cauchy reals and the Dedekind reals are "the same."

Its soundness is validated by the simplicial model of type theory. (The identity type elimination rule remains valid!)

III. Quotients

Quotients

It is in general difficult to form **quotients** in MLTT.

Quotient types:

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma, x : A, y : A \vdash R \text{ type}}{\Gamma \vdash A/R \text{ type}} \qquad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A/R \text{ type}_\ell}{\Gamma \vdash [M] : A/R}$$

$$\frac{\Gamma \vdash M, N : A \qquad \Gamma, x : A, y : A \vdash R \text{ type} \qquad \Gamma \vdash P : R[M, N/x, y]}{\Gamma \vdash \mathsf{Qax}(P) : \mathsf{Id}_{A/R}([M], [N])}$$

and so on... but such types are not necessarily **effective**:

$$\frac{\Gamma, x : A, y : A \vdash R(x, y) \text{ type} \qquad \Gamma \vdash P : \mathsf{Id}_{A/R}([M], [N])}{\Gamma \vdash ???(M, N, P) : R(M, N)}$$

Worse:

Theorem (Maietti 1999)

If quotient types are effective and UIP holds then $A + \neg A$ for small A.

Higher Inductive Types (HITs)

Idea:

$\boxed{\text{When building a type, also specify some paths.}}$

For example, to build a type Int of integers we may postulate:

- for each M : Nat a **positive integer** $\mathsf{pos}(M)$: Int
- for each M : Nat a **negative integer** $\mathsf{neg}(M)$: Int
- an identity $\mathsf{pnZero} : \mathsf{Id}_{\mathsf{Int}}(\mathsf{pos}(\mathsf{zero}), \mathsf{int}(\mathsf{zero}))$

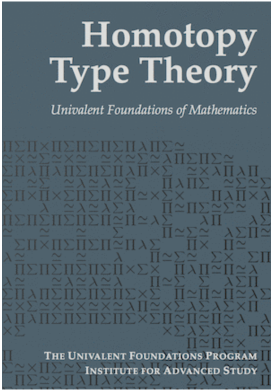
This can be used to specify homotopical spaces synthetically.
E.g. a circle can be specified by postulating

- a base point $\mathsf{base} : \mathbb{S}^1$
- a path $\mathsf{loop} : \mathsf{Id}_{\mathbb{S}^1}(\mathsf{base}, \mathsf{base})$

This leads to **synthetic homotopy theory**. E.g. there is a machine-checked proof that $\pi_4(\mathbb{S}^3) \simeq \mathbb{Z}_2$.

Homotopy Type Theory (HoTT)

The results of Awodey/Warren/Voevodsky led to a flurry of results. This culminated in a Special Year at the IAS in Princeton:



$$\text{HoTT} \stackrel{\text{def}}{=} \text{MLTT} + \text{univalence axiom} + \text{some HITs}$$

IV. Further directions

50 Years of MLTT

Achievements:

- A number of well-behaved type theories…
- …with well-understood semantics.
- One industrial-strength proof assistant: **Coq**.
  Many machine-checked proofs! Greatest hits:
  - Four color theorem [Gonthier 2008]
  - Feit-Thompson odd order theorem [Gonthier et al. 2013]
  - CompCert, a verified C compiler [Leroy et al. 2005–2018]
  - Iris, for verifying concurrent programs [Jung et al. 2018]
- Many 'experimental' proof assistants: Agda, Lean, Arend, …
  Projects to keep an eye on:
  - Kevin Buzzard's Xena Project (in Lean) at Imperial
  - the CMU Hoskinson Center for Formal Mathematics
  - Tim Gowers' project on automated theorem proving (not TT)
- A deep connection between homotopy theory and MLTT.

Where to go from here

Read the HoTT book!

Many directions of work. To name a few:

- **Synthetic homotopy theory**. Better, possibly computational, calculations of homotopy groups of spheres and other spaces.
- **New formalizations of mathematics**. Constructive, machine-checked proofs of known and new results from mathematics.

  Is there some secret higher-dimensional content?
- **Improved or new type theories**. Either adding more power, or improving the computational behaviour of HoTT.
  - cubical type theories
  - modal type theories
  - metatheory, in particular objective metatheory

References

- Steve Awodey, *Structuralism, Invariance, and Univalence*, Categories for the Working Philosopher (Elaine Landry, ed.), Oxford University Press, 2017.
- Martin Hofmann, *Extensional Constructs in Intensional Type Theory*, Distinguished Dissertations, Springer London, 1997.
- Álvaro Pelayo and Michael Warren, *Homotopy type theory and Voevodsky's univalent foundations*, Bulletin of the American Mathematical Society **51** (2014), no. 4, 597–648.
- The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, 2013.