Homotopy Type Theory
and
Univalent Foundations of Mathematics

Steve Awodey
Carnegie Mellon University

February 2012
CMU PAL

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

1. Homotopy can be used as a tool to construct models of systems of logic.

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

1. Homotopy can be used as a tool to construct models of systems of logic.
2. Constructive type theory can be used as a formal calculus to reason about homotopy.

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

1. Homotopy can be used as a tool to construct models of systems of logic.
2. Constructive type theory can be used as a formal calculus to reason about homotopy.
3. The computational implementation of type theory allows computer verified proofs in homotopy theory.

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

1. Homotopy can be used as a tool to construct models of systems of logic.
2. Constructive type theory can be used as a formal calculus to reason about homotopy.
3. The computational implementation of type theory allows computer verified proofs in homotopy theory.
4. The homotopy interpretation suggests new logical constructions and axioms.

Introduction

A new connection has recently come to light between Logic and Topology, namely an interpretation of the constructive type theory of Martin-Löf into homotopy theory.

1. Homotopy can be used as a tool to construct models of systems of logic.
2. Constructive type theory can be used as a formal calculus to reason about homotopy.
3. The computational implementation of type theory allows computer verified proofs in homotopy theory.
4. The homotopy interpretation suggests new logical constructions and axioms.
5. Voevodsky's *Univalent Foundations* program combines these aspects into a new program of foundations for mathematics.

Type theory

Martin-Löf constructive type theory consists of:

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X, \; Y, \ldots, A \times B, \; A \rightarrow B, \ldots$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X,\ Y, \ldots, A \times B,\ A \to B, \ldots$
- **Terms**: $x : A,\ b : B,\ \langle a, b\rangle,\ \lambda x.b(x), \ldots$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X,\ Y, \dots, A \times B,\ A \to B, \dots$
- **Terms**: $x : A,\ b : B,\ \langle a, b \rangle,\ \lambda x.b(x), \dots$
- **Dependent Types**: $x : A \vdash B(x)$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X,\ Y, \ldots, A \times B,\ A \to B, \ldots$
- **Terms**: $x : A,\ b : B,\ \langle a, b \rangle,\ \lambda x.b(x), \ldots$
- **Dependent Types**: $x : A \vdash B(x)$
  - $\sum_{x:A} B(x)$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X,\ Y, \dots, A \times B,\ A \to B, \dots$
- **Terms**: $x : A,\ b : B,\ \langle a, b\rangle,\ \lambda x.b(x), \dots$
- **Dependent Types**: $x : A \vdash B(x)$
  - $\sum_{x:A} B(x)$
  - $\prod_{x:A} B(x)$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X,\ Y, \ldots, A \times B,\ A \to B, \ldots$
- **Terms**: $x : A,\ b : B,\ \langle a, b \rangle,\ \lambda x.b(x), \ldots$
- **Dependent Types**: $x : A \vdash B(x)$
  - $\sum_{x:A} B(x)$
  - $\prod_{x:A} B(x)$
- **Equations** $s = t : A$

Type theory

Martin-Löf constructive type theory consists of:

- **Types**: $X, \ Y, \ldots, A \times B, \ A \to B, \ldots$
- **Terms**: $x : A, \ b : B, \ \langle a, b \rangle, \ \lambda x.b(x), \ldots$
- **Dependent Types**: $x : A \vdash B(x)$
  - $\sum_{x:A} B(x)$
  - $\prod_{x:A} B(x)$
- **Equations** $s = t : A$

Formal calculus of terms and equations – like polynomials, only more complicated.

Propositions as Types

The system has a curious dual interpretation:

- once as **mathematical** objects: types are "sets" and their terms are "elements", which are being constructed,

Propositions as Types

The system has a curious dual interpretation:

- once as **mathematical** objects: types are "sets" and their terms are "elements", which are being constructed,
- once as **logical** objects: types are "propositions" and their terms are "proofs", which are being derived.

Propositions as Types

The system has a curious dual interpretation:

- once as **mathematical** objects: types are "sets" and their terms are "elements", which are being constructed,
- once as **logical** objects: types are "propositions" and their terms are "proofs", which are being derived.

This is also known as the *Curry-Howard correspondence*.

Identity types

According to the logical interpretation we have:

- **propositional logic**: $A \times B, A \to B$,
- **predicate logic**: $B(x), C(x, y)$, with **quantifiers** $\prod$ and $\sum$.

Identity types

According to the logical interpretation we have:

- **propositional logic**: $A \times B, A \to B$,
- **predicate logic**: $B(x), C(x, y)$, with **quantifiers** $\prod$ and $\sum$.

So it's natural to add a primitive relation of **identity** between any terms of the same type:

$$x, y : A \vdash \mathtt{Id}_A(x, y)$$

This type represents the **logical** proposition "x is identical to y".

Identity types

According to the logical interpretation we have:

- **propositional logic**: $A \times B, A \to B$,
- **predicate logic**: $B(x), C(x, y)$, with **quantifiers** $\prod$ and $\sum$.

So it's natural to add a primitive relation of **identity** between any terms of the same type:

$$x, y : A \vdash \mathtt{Id}_A(x, y)$$

This type represents the **logical** proposition "x is identical to y".

On the **mathematical** side, the identity type admits a newly discovered geometric interpretation.

Rules for identity types

The **introduction** rule says that $a : A$ is always identical to itself:

$$\mathtt{r}(a) : \mathtt{Id}_A(a, a)$$

Rules for identity types

The **introduction** rule says that $a : A$ is always identical to itself:

$$\mathtt{r}(a) : \mathtt{Id}_A(a, a)$$

The **elimination** rule is a form of Leibniz's law:

$$\frac{a : A \;\vdash\; d(a) : D\big(a, a, \mathtt{r}(a)\big)}{c : \mathtt{Id}_A(a, b) \;\vdash\; \mathtt{J}_d(a, b, c) : D(a, b, c)}$$

Rules for identity types

The **introduction** rule says that $a : A$ is always identical to itself:

$$\mathtt{r}(a) : \mathtt{Id}_A(a, a)$$

The **elimination** rule is a form of Leibniz's law:

$$\frac{a : A \ \vdash \ d(a) : D\big(a, a, \mathtt{r}(a)\big)}{c : \mathtt{Id}_A(a, b) \ \vdash \ \mathtt{J}_d(a, b, c) : D(a, b, c)}$$

Schematically:

$$D(a, a) \ \& \ \mathtt{Id}_A(a, b) \ \Rightarrow \ D(a, b)$$

Intensionality

The rules are such that if a and b are **equal**:

$$a = b$$

then they are also **identical**:

$$t : \mathtt{Id}_A(a, b) \quad \text{(for some } t\text{)}.$$

Intensionality

The rules are such that if a and b are **equal**:

$$a = b$$

then they are also **identical**:

$$t : \mathtt{Id}_A(a, b) \quad \text{(for some } t\text{).}$$

But the converse is not true — this is called **intensionality**.

Intensionality

The rules are such that if a and b are **equal**:

$$a = b$$

then they are also **identical**:

$$t : \mathtt{Id}_A(a, b) \quad \text{(for some } t\text{)}.$$

But the converse is not true — this is called **intensionality**.

It gives rise to a structure of great combinatorial complexity.

The homotopy interpretation

Suppose we have terms of ascending identity types:

$$\begin{aligned} a,\ b &: A \\ p,\ q &: \mathtt{Id}_A(a,b) \\ \alpha,\ \beta &: \mathtt{Id}_{\mathtt{Id}_A(a,b)}(p,q) \\ \ldots &: \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\ldots}}}(\ldots) \end{aligned}$$

The homotopy interpretation

Suppose we have terms of ascending identity types:

$$\begin{aligned} a,\ b &: A \\ p,\ q &: \mathtt{Id}_A(a,b) \\ \alpha,\ \beta &: \mathtt{Id}_{\mathtt{Id}_A(a,b)}(p,q) \\ \ldots &: \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\ldots}}}(\ldots) \end{aligned}$$

Consider the following interpretation:

$$\begin{array}{rcl} \text{Types} & \rightsquigarrow & \text{Spaces} \\ \text{Terms} & \rightsquigarrow & \text{Maps} \\ a : A & \rightsquigarrow & \text{Points } a : 1 \longrightarrow A \\ p : \mathtt{Id}_A(a,b) & \rightsquigarrow & \text{Paths } p : a \Rightarrow b \\ \alpha : \mathtt{Id}_{\mathtt{Id}_A(a,b)}(p,q) & \rightsquigarrow & \text{Homotopies } \alpha : p \Rrightarrow q \\ \vdots & & \end{array}$$

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.
The identity rules imply the following:

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.
The identity rules imply the following:

$$\frac{p : \mathtt{Id}_A(a, b) \qquad \overline{a} : B(a)}{p_* \overline{a} : B(b)}$$

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.
The identity rules imply the following:

$$\frac{p : \mathtt{Id}_A(a, b) \qquad \bar{a} : B(a)}{p_* \bar{a} : B(b)}$$

Logically, this just says "$a = b$ & $B(a) \Rightarrow B(b)$".

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.
The identity rules imply the following:

$$\frac{p : \mathtt{Id}_A(a, b) \qquad \overline{a} : B(a)}{p_* \overline{a} : B(b)}$$

Logically, this just says "$a = b \ \& \ B(a) \Rightarrow B(b)$".

But topologically, it is a **lifting property**:

$$\begin{tikzcd}
B \arrow[d] & \overline{a} \arrow[r, dotted] & p_* \overline{a} \\
A & a \arrow[r, "p"'] & b
\end{tikzcd}$$

The homotopy interpretation

We still need to interpret dependent types $x : A \vdash B(x)$.
The identity rules imply the following:

$$\frac{p : \mathtt{Id}_A(a, b) \qquad \bar{a} : B(a)}{p_* \bar{a} : B(b)}$$

Logically, this just says "$a = b \;\&\; B(a) \Rightarrow B(b)$".

But topologically, it is a **lifting property**:

$$\begin{tikzcd} B \arrow[d] & \bar{a} \arrow[r, dotted] & p_* \bar{a} \\ A & a \arrow[r, "p"'] & b \end{tikzcd}$$

This is the notion of a "fibration".

The homotopy interpretation

Thus we continue the homotopy interpretation as follows:

$$\text{Dependent types} \quad x : A \vdash B(x) \quad \rightsquigarrow \quad \text{Fibrations} \quad \begin{tikzcd} B \arrow[d] \\ A \end{tikzcd}$$

The homotopy interpretation

Thus we continue the homotopy interpretation as follows:

$$\text{Dependent types} \quad x : A \vdash B(x) \quad \rightsquigarrow \quad \text{Fibrations} \quad \begin{array}{c} B \\ \big\downarrow \\ A \end{array}$$

The type $B(a)$ is the fiber of $B \longrightarrow A$ over the point $a : A$

$$\begin{tikzcd} B(a) \arrow[r] \arrow[d] & B \arrow[d] \\ 1 \arrow[r, "a"'] & A. \end{tikzcd}$$

The homotopy interpretation

To interpret the identity type $x, y : A \vdash \mathtt{Id}_A(x, y)$, we thus require a fibration over $A \times A$.

The homotopy interpretation

To interpret the identity type $x, y : A \vdash \mathtt{Id}_A(x, y)$, we thus require a fibration over $A \times A$.
Take the space A^I of all paths in A:

Identity type $\quad x, y : A \vdash \mathtt{Id}_A(x, y) \quad \rightsquigarrow \quad$ Path space $\quad$

$$\begin{tikzcd} A^I \arrow[d] \\ A \times A \end{tikzcd}$$

The homotopy interpretation

To interpret the identity type $x, y : A \vdash \mathtt{Id}_A(x, y)$, we thus require a fibration over $A \times A$.
Take the space A^I of all paths in A:

$$\text{Identity type} \quad x, y : A \vdash \mathtt{Id}_A(x, y) \quad \rightsquigarrow \quad \text{Path space} \quad \begin{array}{c} A^I \\ \downarrow \\ A \times A \end{array}$$

The fiber $\mathtt{Id}_A(a, b)$ over a point $(a, b) \in A \times A$ is the space of paths from a to b in A.

$$\begin{tikzcd}
\mathtt{Id}_A(a,b) \arrow[r] \arrow[d] & A^I \arrow[d] \\
1 \arrow[r, "{(a,b)}"'] & A \times A.
\end{tikzcd}$$

The homotopy interpretation

The path space A^I classifies homotopies $\vartheta : f \Rightarrow g$ between maps $f, g : X \to A$,

$$\begin{tikzcd} & A^I \arrow[d] \\ X \arrow[ur, "\vartheta"] \arrow[r, "{(f,g)}"'] & A \times A. \end{tikzcd}$$

The homotopy interpretation

The path space A^I classifies homotopies $\vartheta : f \Rightarrow g$ between maps $f, g : X \to A$,

$$\begin{tikzcd} & A^I \arrow[d] \\ X \arrow[ur, "\vartheta"] \arrow[r, "(f,g)"'] & A \times A. \end{tikzcd}$$

So given any terms $x : X \vdash f, g : A$, an identity term

$$x : X \vdash \vartheta : \mathtt{Id}_A(f, g)$$

is interpreted as a homotopy between f and g.

The homotopy interpretation

Instead of concrete spaces and homotopies, for the formal interpretation we use the abstract axiomatic description provided by **Quillen model categories**.

The homotopy interpretation

Instead of concrete spaces and homotopies, for the formal interpretation we use the abstract axiomatic description provided by **Quillen model categories**.

- Gives a wide range of different models.

The homotopy interpretation

Instead of concrete spaces and homotopies, for the formal interpretation we use the abstract axiomatic description provided by **Quillen model categories**.

- Gives a wide range of different models.
- Includes classical homotopy of spaces and simplicial sets.

The homotopy interpretation

Instead of concrete spaces and homotopies, for the formal interpretation we use the abstract axiomatic description provided by **Quillen model categories**.

- Gives a wide range of different models.
- Includes classical homotopy of spaces and simplicial sets.
- Allows the use of standard methods from categorical logic.

Soundness of the homotopy interpretation

Theorem (Awodey & Warren 2008)

Martin-Löf type theory has a **sound** *interpretation into any Quillen model category.*

Soundness of the homotopy interpretation

Theorem (Awodey & Warren 2008)

Martin-Löf type theory has a **sound** *interpretation into any Quillen model category.*

Remarks.

- We consider here only the "theory of identity", no $\sum$ or $\prod$.

Soundness of the homotopy interpretation

Theorem (Awodey & Warren 2008)

Martin-Löf type theory has a **sound** *interpretation into any Quillen model category.*

Remarks.

- We consider here only the "theory of identity", no $\sum$ or $\prod$.
- There is an issue of "coherence" of the interpretation, which requires a technical condition on the QMC.

Soundness of the homotopy interpretation

Theorem (Awodey & Warren 2008)

Martin-Löf type theory has a **sound** *interpretation into any Quillen model category.*

Remarks.

- We consider here only the "theory of identity", no $\sum$ or $\prod$.
- There is an issue of "coherence" of the interpretation, which requires a technical condition on the QMC.
- One doesn't need the full QMC structure, but only a *weak factorization system*.

Soundness and completeness

The logical notion of **soundness** means that a provable statement is always true under the specified interpretation:

$$\text{provable} \quad \xrightarrow{\textit{sound}} \quad \text{true in all models}$$

Soundness and completeness

The logical notion of **soundness** means that a provable statement is always true under the specified interpretation:

$$\text{provable} \quad \xRightarrow{\textit{sound}} \quad \text{true in all models}$$

The converse notion is **completeness**: a statement is provable if its interpretation is always true:

$$\text{provable} \quad \xLeftarrow{\textit{complete}} \quad \text{true in all models}$$

Completeness of the homotopy interpretation

Theorem (Gambino & Garner 2009)

The homotopy interpretation of Martin-Löf type theory is also **complete**.

Completeness of the homotopy interpretation

Theorem (Gambino & Garner 2009)

The homotopy interpretation of Martin-Löf type theory is also **complete***.*

More precisely: in the theory of identity, a statement that is true under any coherent interpretation in a weak factorization system is also provable.

Completeness of the homotopy interpretation

Theorem (Gambino & Garner 2009)

The homotopy interpretation of Martin-Löf type theory is also **complete**.

More precisely: in the theory of identity, a statement that is true under any coherent interpretation in a weak factorization system is also provable.

A benefit of the abstract semantics: the proof uses the standard method of *syntactic categories* to construct a canonical model.

Conclusion of Part I

Martin-Löf type theory provides a "logic of homotopy".

The Fundamental Groupoid of a Type

It's now reasonable to ask, how **expressive** is the logical system as a formal language for homotopy theory?

The Fundamental Groupoid of a Type

It's now reasonable to ask, how **expressive** is the logical system as a formal language for homotopy theory?

What homotopically relevant facts, properties, and constructions are logically expressible?

The Fundamental Groupoid of a Type

It's now reasonable to ask, how **expressive** is the logical system as a formal language for homotopy theory?

What homotopically relevant facts, properties, and constructions are logically expressible?

One example: the topological **fundamental group** and its higher generalizations are logical constructions.

Fundamental groupoids

Let's return to the system of identity terms of various orders:

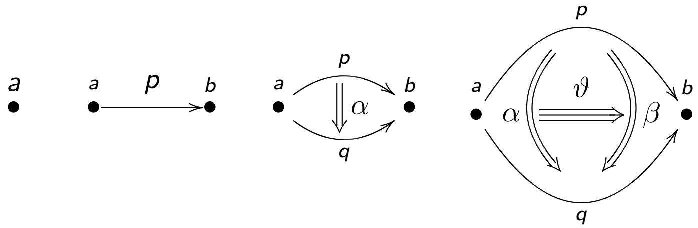
$$\begin{aligned} a,\ b &: A \\ p,\ q &: \mathtt{Id}_A(a,b) \\ \alpha,\ \beta &: \mathtt{Id}_{\mathtt{Id}_A(a,b)}(p,q) \\ \vartheta &: \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\dots}}}(\alpha,\beta) \end{aligned}$$

Fundamental groupoids

Let's return to the system of identity terms of various orders:

$$\begin{aligned} a,\ b &: A \\ p,\ q &: \mathtt{Id}_A(a,b) \\ \alpha,\ \beta &: \mathtt{Id}_{\mathtt{Id}_A(a,b)}(p,q) \\ \vartheta &: \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\dots}}}(\alpha,\beta) \end{aligned}$$

These can be represented suggestively as follows:



Fundamental groupoids

As in topology, the terms of order 0 and 1, (“points” and “paths”),

$$a \xrightarrow{p} b$$

bear the structure of a **groupoid**.

Fundamental groupoids

As in topology, the terms of order 0 and 1, ("points" and "paths"),

$$a \xrightarrow{p} b$$

bear the structure of a **groupoid**.

The laws of identity correspond to the **groupoid operations**:

$$\begin{array}{rll}
r : \mathtt{Id}(a, a) & \text{reflexivity} & a \longrightarrow a \\
s : \mathtt{Id}(a, b) \longrightarrow \mathtt{Id}(b, a) & \text{symmetry} & a \leftrightarrows b \\
t : \mathtt{Id}(a, b) \times \mathtt{Id}(b, c) \longrightarrow \mathtt{Id}(a, c) & \text{transitivity} & a \longrightarrow b \longrightarrow c
\end{array}$$

Fundamental groupoids

As in topology, the terms of order 0 and 1, ("points" and "paths"),

$$a \xrightarrow{p} b$$

bear the structure of a **groupoid**.

The laws of identity correspond to the **groupoid operations**:

$$\begin{array}{rll}
r : \mathtt{Id}(a, a) & \text{reflexivity} & a \longrightarrow a \\
s : \mathtt{Id}(a, b) \longrightarrow \mathtt{Id}(b, a) & \text{symmetry} & a \leftrightarrows b \\
t : \mathtt{Id}(a, b) \times \mathtt{Id}(b, c) \longrightarrow \mathtt{Id}(a, c) & \text{transitivity} & a \longrightarrow b \longrightarrow c
\end{array}$$

This was first shown by Hofmann & Streicher (1998), who gave a model of intensional type theory using groupoids as types.

Fundamental groupoids

But also just as in topology, the **groupoid equations** of associativity, inverse, and unit:

$$p \cdot (q \cdot r) = (p \cdot q) \cdot r$$
$$p^{-1} \cdot p = 1 = p \cdot p^{-1}$$
$$1 \cdot p = p = p \cdot 1$$

do not hold strictly, but only "up to homotopy".

Fundamental groupoids

But also just as in topology, the **groupoid equations** of associativity, inverse, and unit:

$$p \cdot (q \cdot r) = (p \cdot q) \cdot r$$
$$p^{-1} \cdot p = 1 = p \cdot p^{-1}$$
$$1 \cdot p = p = p \cdot 1$$

do not hold strictly, but only "up to homotopy".

This means they are witnessed by terms of the next higher order:

$$\vartheta : \mathtt{Id}_{\mathtt{Id}}\left(p^{-1} \cdot p, 1\right)$$

$$\begin{tikzcd}
 & b \arrow[dr, bend left, "p^{-1}"] \arrow[d, phantom, "\Downarrow \vartheta"] & \\
a \arrow[ur, bend left, "p"] \arrow[rr, bend right, "1"'] & \, & a
\end{tikzcd}$$

Fundamental groupoids

The entire system of identity terms of all orders forms an infinite-dimensional graph, or "globular set":

$$A \leftleftarrows \mathtt{Id}_A \leftleftarrows \mathtt{Id}_{\mathtt{Id}_A} \leftleftarrows \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A}} \leftleftarrows \dots$$

Fundamental groupoids

The entire system of identity terms of all orders forms an infinite-dimensional graph, or "globular set":

$$A \leftleftarrows \mathtt{Id}_A \leftleftarrows \mathtt{Id}_{\mathtt{Id}_A} \leftleftarrows \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A}} \leftleftarrows \dots$$

It has the structure of a **weak**, infinite-dimensional, groupoid (as defined by Batanin 1998 and occurring homotopy theory):

Theorem (Lumsdaine, Garner & van den Berg, 2009)

The system of identity terms of all orders over any fixed type is a weak ω-groupoid.

Fundamental groupoids

The entire system of identity terms of all orders forms an infinite-dimensional graph, or "globular set":

$$A \leftleftarrows \mathtt{Id}_A \leftleftarrows \mathtt{Id}_{\mathtt{Id}_A} \leftleftarrows \mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_A}} \leftleftarrows \ldots$$

It has the structure of a **weak**, infinite-dimensional, groupoid (as defined by Batanin 1998 and occurring homotopy theory):

Theorem (Lumsdaine, Garner & van den Berg, 2009)

The system of identity terms of all orders over any fixed type is a weak ω-groupoid.

Every type has a **fundamental groupoid**.

Conclusion of Part II

- The fundamental groupoid of a *space* is a **logical** construction.

Conclusion of Part II

- The fundamental groupoid of a *space* is a **logical** construction.
- The topological fact that points, paths, and (higher) homotopies form a weak, higher dimensional groupoid, is not just analogous to type theory; it's the same construction.

Conclusion of Part II

- The fundamental groupoid of a *space* is a **logical** construction.
- The topological fact that points, paths, and (higher) homotopies form a weak, higher dimensional groupoid, is not just analogous to type theory; it's the same construction.
- Grothendieck's "Homotopy Hypothesis": weak ω-groupoids classify homotopy types of spaces.

Conclusion of Part II

- The fundamental groupoid of a *space* is a **logical** construction.
- The topological fact that points, paths, and (higher) homotopies form a weak, higher dimensional groupoid, is not just analogous to type theory; it's the same construction.
- Grothendieck's "Homotopy Hypothesis": weak ω-groupoids classify homotopy types of spaces.
- Logical methods suffice in principle to capture a great deal of homotopy theory.

Univalent Foundations

Vladimir Voevodsky's *Univalent Foundations Program* combines:

Univalent Foundations

Vladimir Voevodsky's *Univalent Foundations Program* combines:

- the foregoing representation of homotopy theory in constructive type theory

Univalent Foundations

Vladimir Voevodsky's *Univalent Foundations Program* combines:

- the foregoing representation of homotopy theory in constructive type theory
- the well-developed implementations of type theory in computational proof assistants like Coq and Agda.

Univalent Foundations

Vladimir Voevodsky's *Univalent Foundations Program* combines:

- the foregoing representation of homotopy theory in constructive type theory
- the well-developed implementations of type theory in computational proof assistants like Coq and Agda.

Allows computer verified proofs in homotopy theory, and related fields.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

- the fundamental group $\pi_1(X, b)$ of a type X at basepoint $b : X$ consists of terms of type $\mathtt{Id}_X(b, b)$.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

- the fundamental group $\pi_1(X, b)$ of a type X at basepoint $b : X$ consists of terms of type $\mathtt{Id}_X(b, b)$.
- the second homotopy group $\pi_2(X, b)$ consists of terms of type $\mathtt{Id}_{\mathtt{Id}_X(b,b)}(\mathtt{r}(b), \mathtt{r}(b))$.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

- the fundamental group $\pi_1(X, b)$ of a type X at basepoint $b : X$ consists of terms of type $\mathtt{Id}_X(b, b)$.
- the second homotopy group $\pi_2(X, b)$ consists of terms of type $\mathtt{Id}_{\mathtt{Id}_X(b,b)}(\mathtt{r}(b), \mathtt{r}(b))$.
- Each of these types has a group structure, and so the second one has *two* group structures that are compatible.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

- the fundamental group $\pi_1(X, b)$ of a type X at basepoint $b : X$ consists of terms of type $\mathtt{Id}_X(b, b)$.
- the second homotopy group $\pi_2(X, b)$ consists of terms of type $\mathtt{Id}_{\mathtt{Id}_X(b,b)}(\mathtt{r}(b), \mathtt{r}(b))$.
- Each of these types has a group structure, and so the second one has *two* group structures that are compatible.
- Now the Eckmann-Hilton argument shows that the two structures on $\pi_2(X, b)$ agree, and are abelian.

A computational example

A classical result states that the higher homotopy groups of a space are always abelian.
We can formalize this in type theory:

- the fundamental group $\pi_1(X, b)$ of a type X at basepoint $b : X$ consists of terms of type $\mathtt{Id}_X(b, b)$.
- the second homotopy group $\pi_2(X, b)$ consists of terms of type $\mathtt{Id}_{\mathtt{Id}_X(b,b)}(\mathtt{r}(b), \mathtt{r}(b))$.
- Each of these types has a group structure, and so the second one has *two* group structures that are compatible.
- Now the Eckmann-Hilton argument shows that the two structures on $\pi_2(X, b)$ agree, and are abelian.

This argument can be formalized in the automated proof assistant Coq and verified to be correct. In this way, we can use the homotopical interpretation to verify proofs in homotopy theory.

A computational example

```
(* An adaptation to Coq of Dan Licata's Agda proof that the higher homotopy groups are abelian,
   by Jeremy Avigad. This file depends on the library in the "UnivalentFoundations" directory of
   Andrej Bauer's Github repository. The code is written for Coq 8.3, which means that variables
   are introduced automatically.*)

Implicit Arguments homotopy_concat [A x y z p p' q q'].
Implicit Arguments idpath_left_unit [A x y].
Implicit Arguments idpath_right_unit [A x y].

Lemma map2 {A B C} {x x' : A} {y y' : B} (f : A -> B -> C)
  (p : x ~~> x') (p' : y ~~> y') : (f x y) ~~> (f x' y').
Proof. induction p; induction p'; trivial. Defined.

(* The next four lemmas are needed to prove the left and right identity laws,
    generalizing those laws to path spaces. *)

Lemma adjust_l {A} {x y : A} {p q : x ~~> y} (R : p ~~> q) :
  idpath x @ p ~~> idpath x @ q.
Proof. exact (idpath_left_unit p @ R @ !(idpath_left_unit q)). Defined.
(* induction R doesn't given a term that is explicit enough. *)

Lemma homotopy_concat_id_left {A} {x y : A} {p p' : x ~~> y}
  (R : p ~~> p') : homotopy_concat (idpath (idpath x)) R ~~> adjust_l R.
Proof. induction R; induction x0; trivial. Defined.

Lemma adjust_r {A} {x y: A} {p q : x ~~> y} (R : p ~~> q) :
  p @ idpath y ~~> q @ idpath y.
Proof. exact (idpath_right_unit p @ R @ !(idpath_right_unit q)). Defined.
```

```
Lemma homotopy_concat_id_right {A} {x y : A} {p p' : x ~~> y}
 (R : p ~~> p') : homotopy_concat R (idpath (idpath y)) ~~> adjust_r R.
Proof. induction R; induction x0; trivial. Defined.

Lemma concat_interchange {A} {x y z : A} {p q r : x ~~> y} {p' q' r' : y ~~> z}
  {R : p ~~> q} {S : q ~~> r} {T : p' ~~> q'} {U : q' ~~> r'} :
  homotopy_concat (R @ S) (T @ U) ~~>
    (homotopy_concat R T) @ (homotopy_concat S U).
Proof.
  induction R; induction S; induction T; induction U.
  induction x0; induction x1; trivial.
Defined.

(* Here is the standard proof. It is phrased in terms of Pi_2, but instantiating "A" and "base"
   accordingly yields the corresponding result for any n >= 2. *)

Section Pi2_Abelian.

Variables (A : Type) (base : A).

Definition Pi1 := (base ~~> base).
Definition Pi2 := (idpath base) ~~> (idpath base).

Notation "p @@ q" := (homotopy_concat p q) (at level 60).
Notation "[id]" := (idpath (idpath base)).

Lemma comp_left_unit {p : Pi2} : [id] @@ p ~~> p.
Proof.
  apply (concat (homotopy_concat_id_left p)).
  path_via (p @ [id]); apply idpath_left_unit.
Defined.
```

```
Lemma comp_right_unit {p : Pi2} : p @@ [id] ~~> p.
  apply (concat (homotopy_concat_id_right p)).
  path_via (p @ [id]); apply idpath_left_unit.
Defined.

Lemma comp_interchange  {a b c d : Pi2} :
  (a @ b) @@ (c @ d) ~~> (a @@ c) @ (b @@ d).
Proof. exact concat_interchange. Defined.

Lemma comp_same {a b : Pi2} : a @ b ~~> a @@ b.
Proof.
  path_via ((a @@ [id]) @ b). apply (!comp_right_unit).
  path_via ((a @@ [id]) @ ([id] @@ b)). apply (!comp_left_unit).
  path_via ((a @ [id]) @@ ([id] @ b)). apply (!comp_interchange).
  path_via (a @@ ([id] @ b)).
    apply map2; [apply idpath_right_unit | apply idpath].
  apply map2; [apply idpath | apply idpath_left_unit].
Defined.

(* Here path_via calls path_tricks, which decomposes "_ @ _ = _ @ _"  too aggressively. *)
Lemma Pi2_abelian {a b : Pi2} : a @ b ~~> b @ a.
Proof.
  apply @concat with (y := ([id] @@ a) @ b).
    path_tricks; apply (!comp_left_unit).
  apply @concat with (y := ([id] @@ a) @ (b @@ [id])).
    path_tricks; apply (!comp_right_unit).
  apply (concat (!comp_interchange)); apply (concat (!comp_same)); path_tricks.
Defined.

End Pi2_Abelian.
```

Conclusion of Part III

- Voevodsky has implemented a large amount of basic homotopy theory, and proven some surprising new results in foundations.

Conclusion of Part III

- Voevodsky has implemented a large amount of basic homotopy theory, and proven some surprising new results in foundations.
- The program is being pursued by a group of researchers, formalizing parts of homotopy theory and other mathematics in this settting.

Conclusion of Part III

- Voevodsky has implemented a large amount of basic homotopy theory, and proven some surprising new results in foundations.
- The program is being pursued by a group of researchers, formalizing parts of homotopy theory and other mathematics in this settting.
- Some new logical constructions and axioms are suggested by the homotopy interpretation.

Higher-dimensional inductive types

(Work in progress by Lumsdaine, Shulmann & others.)

Higher-dimensional inductive types

(Work in progress by Lumsdaine, Shulmann & others.)

The natural numbers $\mathbb{N}$ are implemented in type theory as an inductively defined structure of type:

$$o : \mathbb{N}$$
$$s : \mathbb{N} \to \mathbb{N}$$

Higher-dimensional inductive types

(Work in progress by Lumsdaine, Shulmann & others.)

The natural numbers $\mathbb{N}$ are implemented in type theory as an inductively defined structure of type:

$$o : \mathbb{N}$$
$$s : \mathbb{N} \to \mathbb{N}$$

The recursion property is captured by an elimination rule:

$$\frac{a : X \qquad f : X \to X}{\mathrm{rec}(a, f) : \mathbb{N} \to X}$$

Higher-dimensional inductive types

(Work in progress by Lumsdaine, Shulmann & others.)

The natural numbers $\mathbb{N}$ are implemented in type theory as an inductively defined structure of type:

$$o : \mathbb{N}$$
$$s : \mathbb{N} \to \mathbb{N}$$

The recursion property is captured by an elimination rule:

$$\frac{a : X \qquad f : X \to X}{\mathrm{rec}(a, f) : \mathbb{N} \to X}$$

such that:

$$\mathrm{rec}(a, f)(o) = a$$
$$\mathrm{rec}(a, f)(sn) = f(\mathrm{rec}(a, f)(n))$$

Higher-dimensional inductive types

This says just that $(\mathbb{N}, o, s)$ is the *free* structure of this type:

$$\begin{tikzcd} & 1 \arrow[dl, "o"'] \arrow[dr, "a"] & \\ \mathbb{N} \arrow["s"', loop left] \arrow[rr, dotted, "\mathsf{rec}"'] & & X \arrow["f", loop right] \end{tikzcd}$$

Higher-dimensional inductive types

This says just that $(\mathbb{N}, o, s)$ is the *free* structure of this type:

$$\begin{tikzcd} & 1 \arrow[dl, "o"'] \arrow[dr, "a"] & \\ \mathbb{N} \arrow[loop left, "s"] \arrow[rr, dotted, "\mathsf{rec}"'] & & X \arrow[loop right, "f"] \end{tikzcd}$$

The map $\mathsf{rec}(a, f) : \mathbb{N} \to X$ is unique with this property.

Higher-dimensional inductive types

The topological circle $\mathbb{S} = S^1$ can also be given as an inductive type, now involving a higher-dimensional generator:

$$b : \mathbb{S}$$
$$p : b \rightsquigarrow b$$

Here we have written $p : b \rightsquigarrow b$ for the "loop" $p : \mathtt{Id}_{\mathbb{S}}(b, b)$.

Higher-dimensional inductive types

There is an associated recursion property, captured again by an elimination rule:

$$\frac{a : X \qquad q : a \rightsquigarrow a}{\mathrm{rec}(a, q) : \mathbb{S} \to X}$$

Higher-dimensional inductive types

There is an associated recursion property, captured again by an elimination rule:

$$\frac{a : X \qquad q : a \rightsquigarrow a}{\mathrm{rec}(a, q) : \mathbb{S} \to X}$$

such that:

$$\mathrm{rec}(a, q)(b) = a$$
$$\mathrm{rec}(a, q)_1(p) = q$$

Here $\mathrm{rec}(a, q)_1$ is the effect of the map $\mathrm{rec}(a, q)$ on paths.

Higher-dimensional inductive types

This says that $(\mathbb{S}, b, p)$ is the *free* structure of this (higher) type:

$$\begin{tikzcd} & & & 1 \arrow[dl, "b"'] \arrow[dr, "a"] & & \\ b \arrow[loop left, "p"] & & \mathbb{S} \arrow[rr, dotted, "\mathsf{rec}"'] & & X & a \arrow[loop right, "q"] \end{tikzcd}$$

Higher-dimensional inductive types

This says that $(\mathbb{S}, b, p)$ is the *free* structure of this (higher) type:

$$
\begin{tikzcd}
 & & & 1 \arrow[dl, "b"'] \arrow[dr, "a"] & & & \\
b \arrow[loop left, "p"] & & \mathbb{S} \arrow[rr, dotted, "\mathsf{rec}"'] & & X & & a \arrow[loop right, "q"]
\end{tikzcd}
$$

The map $\mathsf{rec}(a, q) : \mathbb{S} \to X$ is then unique up to homotopy.

Higher-dimensional inductive types

Here is a sanity check:

Theorem (Shulmann 2011)

The type-theoretic circle $\mathbb{S}$ has the correct homotopy groups: $\pi_1(\mathbb{S}) = \mathbb{Z}$, and $\pi_n(\mathbb{S}) = 0$ when $n \neq 1$.

Higher-dimensional inductive types

Here is a sanity check:

Theorem (Shulmann 2011)

The type-theoretic circle $\mathbb{S}$ *has the correct homotopy groups:* $\pi_1(\mathbb{S}) = \mathbb{Z}$, *and* $\pi_n(\mathbb{S}) = 0$ *when* $n \neq 1$.

The proof is implemented in Coq. It combines classical homotopy theory with methods from constructive type theory, and uses Voevodsky's new Univalence Axiom.

Higher-dimensional inductive types

The unit interval $\mathbb{I} = [0, 1]$ is also an inductive type, on the data:

$$\begin{aligned} 0, 1 &: \mathbb{I} \\ p &: 0 \rightsquigarrow 1 \end{aligned}$$

Again we have written $p : 0 \rightsquigarrow 1$ for the path $p : \mathtt{Id}_{\mathbb{I}}(0, 1)$.

Higher-dimensional inductive types

The unit interval $\mathbb{I} = [0, 1]$ is also an inductive type, on the data:

$$\begin{aligned} 0, 1 &: \mathbb{I} \\ p &: 0 \rightsquigarrow 1 \end{aligned}$$

Again we have written $p : 0 \rightsquigarrow 1$ for the path $p : \mathtt{Id}_{\mathbb{I}}(0, 1)$.

Remark. In topology, the interval is used to define the notion of a path. Here we have the notion of a path as a logical primitive, and can use it to define the interval.

Higher-dimensional inductive types

Many other basic spaces and constructions can be introduced in this way:

- the higher spheres S^n and disks D^n,
- the suspension ΣA of a space A,
- finite cell complexes, tori, cylinders, ...,
- homotopy algebras – i.e. algebraic structures with equations holding up to homotopy,
- the mapping cylinder of a map $f : A \to B$.

Higher-dimensional inductive types

Many other basic spaces and constructions can be introduced in this way:

- the higher spheres S^n and disks D^n,
- the suspension ΣA of a space A,
- finite cell complexes, tori, cylinders, ...,
- homotopy algebras – i.e. algebraic structures with equations holding up to homotopy,
- the mapping cylinder of a map $f : A \to B$.

Using higher-inductive types, one can show there is a rudimentary Quillen model structure in the type theory.

Some further topics

- The Univalence Axiom:

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

- Consistency:

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

- Consistency:

  Voevodsky has constructed a model of the Univalence Axiom in simplicial sets.

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

- Consistency:

  Voevodsky has constructed a model of the Univalence Axiom in simplicial sets.

- Foundations:

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

- Consistency:

  Voevodsky has constructed a model of the Univalence Axiom in simplicial sets.

- Foundations:

  Sets are spaces too, as are propositions, so logic and set theory are subsumed.

Some further topics

- The Univalence Axiom:

  Isomorphic structures are identical.

- Consistency:

  Voevodsky has constructed a model of the Univalence Axiom in simplicial sets.

- Foundations:

  Sets are spaces too, as are propositions, so logic and set theory are subsumed.

New "invariant" foundations for mathematics, with geometric content and a computational character.

Conclusion

Under this new homotopy interpretation, constructive type theory captures a substantial amount of homotopy theory, permitting purely formal reasoning which can even be implemented on a computer.

Conclusion

Under this new homotopy interpretation, constructive type theory captures a substantial amount of homotopy theory, permitting purely formal reasoning which can even be implemented on a computer.

The homotopy interpretation also suggests a new approach to foundations of math with intrinsic geometric content, capturing some forms of reasoning more naturally than traditional foundations in set theory.

References and Further Information

www.HomotopyTypeTheory.org