The Philosophical Logic of Homotopy Type Theory

James Ladyman

University of Bristol

September 8, 2017

Acknowledgments

This talk is based on my work with Stuart Presnell on our project **"Applying Homotopy Type Theory in Logic, Metaphysics, and Philosophy of Physics"**, funded by Leverhulme Trust research project grant RPG-2013-228.

http://bristol.ac.uk/homotopy-type-theory

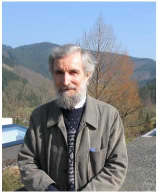
or search for *bristol homotopy type theory*

- Homotopy Type Theory is a research programme in mathematics, founded by (among others) Vladimir Voevodsky and Steve Awodey, building on the work of Per Martin-Löf.

Introduction

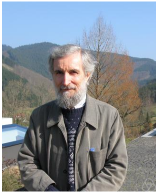
- Homotopy Type Theory is a research programme in mathematics, founded by (among others) Vladimir Voevodsky and Steve Awodey, building on the work of Per Martin-Löf.





- Homotopy Type Theory is a research programme in mathematics, founded by (among others) Vladimir Voevodsky and Steve Awodey, building on the work of Per Martin-Löf.

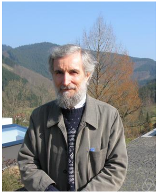




- Aside from its interest as an area of mathematics connecting algebraic topology with logic and computer science, it has also been proposed as a new foundation for mathematical practice.

Introduction

- Homotopy Type Theory is a research programme in mathematics, founded by (among others) Vladimir Voevodsky and Steve Awodey, building on the work of Per Martin-Löf.

- Aside from its interest as an area of mathematics connecting algebraic topology with logic and computer science, it has also been proposed as a new foundation for mathematical practice.
- It is based on constructive intensional dependent type theory, not on ZFC set theory or category theory.

The 'Origin Story' of HoTT

- In 1989 Mikhail Kapranov and Vladimir Voevodsky published a paper relating homotopy theory and groupoids.

The 'Origin Story' of HoTT

- In 1989 Mikhail Kapranov and Vladimir Voevodsky published a paper relating homotopy theory and groupoids.
- In 1998 Carlos Simpson produced a paper claiming to have a counterexample to the main result of K&V's paper.

THE 'ORIGIN STORY' OF HOTT

- In 1989 Mikhail Kapranov and Vladimir Voevodsky published a paper relating homotopy theory and groupoids.
- In 1998 Carlos Simpson produced a paper claiming to have a counterexample to the main result of K&V's paper.
- Voevodsky: "Simpson claimed to have constructed a counterexample, but he was not able to show where the mistake was in our paper. Because of this, it was not clear whether we made a mistake somewhere in our paper or he made a mistake somewhere in his counterexample."

The 'Origin Story' of HoTT

Voevodsky:

- "In 1999–2000 [. . .] I was giving a series of lectures, and Pierre Deligne was taking notes and checking every step of my arguments. Only then did I discover that the proof of a key lemma in my [1993] paper contained a mistake and that the lemma, as stated, could not be salvaged."

The 'Origin Story' of HoTT

Voevodsky:

- "In 1999–2000 [. . .] I was giving a series of lectures, and Pierre Deligne was taking notes and checking every step of my arguments. Only then did I discover that the proof of a key lemma in my [1993] paper contained a mistake and that the lemma, as stated, could not be salvaged."
- "[Since] 1993, multiple groups of mathematicians studied my paper at seminars and used it in their work and none of them noticed the mistake."

The 'Origin Story' of HoTT

Voevodsky:

- "In 1999–2000 [. . .] I was giving a series of lectures, and Pierre Deligne was taking notes and checking every step of my arguments. Only then did I discover that the proof of a key lemma in my [1993] paper contained a mistake and that the lemma, as stated, could not be salvaged."
- "[Since] 1993, multiple groups of mathematicians studied my paper at seminars and used it in their work and none of them noticed the mistake."
- "And it clearly was not an accident. A technical argument by a trusted author, which is hard to check and looks similar to arguments known to be correct, is hardly ever checked in detail."

The 'Origin Story' of HoTT

Voevodsky:

- "In 1999–2000 [. . .] I was giving a series of lectures, and Pierre Deligne was taking notes and checking every step of my arguments. Only then did I discover that the proof of a key lemma in my [1993] paper contained a mistake and that the lemma, as stated, could not be salvaged."
- "[Since] 1993, multiple groups of mathematicians studied my paper at seminars and used it in their work and none of them noticed the mistake."
- "And it clearly was not an accident. A technical argument by a trusted author, which is hard to check and looks similar to arguments known to be correct, is hardly ever checked in detail."
- "**This story got me scared.**"

The 'Origin Story' of HoTT

Voevodsky:

- "It soon became clear that the only long-term solution was somehow to make it possible for me to use computers to verify my abstract, logical, and mathematical constructions."

The 'Origin Story' of HoTT

Voevodsky:

- "It soon became clear that the only long-term solution was somehow to make it possible for me to use computers to verify my abstract, logical, and mathematical constructions."
- "There were several groups developing such systems, but none of them was in any way appropriate for the kind of mathematics for which I needed a system."

Voevodsky:

- "It soon became clear that the only long-term solution was somehow to make it possible for me to use computers to verify my abstract, logical, and mathematical constructions."
- "There were several groups developing such systems, but none of them was in any way appropriate for the kind of mathematics for which I needed a system."
- "The primary challenge that needed to be addressed was that *the foundations of mathematics were unprepared for the requirements of the task*."

- Homotopy Type Theory/Univalent Foundations is first and foremost a research programme connecting algebraic topology with computer science, but it has also been proposed as a 'foundation' for mathematics.

- Homotopy Type Theory/Univalent Foundations is first and foremost a research programme connecting algebraic topology with computer science, but it has also been proposed as a 'foundation' for mathematics.
- In this talk I will explore some of the philosophically interesting features of HoTT/UF.

What is HoTT?

- Homotopy Type Theory/Univalent Foundations as presented in the 'HoTT Book' is built on the constructive intensional type theory of Martin-Löf (MLTT).

What is HoTT?

- Homotopy Type Theory/Univalent Foundations as presented in the ‘HoTT Book’ is built on the constructive intensional type theory of Martin-Löf (MLTT).
- However, there are other approaches in the research programme.

What is HoTT?

- Homotopy Type Theory/Univalent Foundations as presented in the 'HoTT Book' is built on the constructive intensional type theory of Martin-Löf (MLTT).
- However, there are other approaches in the research programme.
- It is not so easy to say what the essential features of HoTT/UF are.

WHAT IS HoTT?

- Homotopy Type Theory/Univalent Foundations as presented in the 'HoTT Book' is built on the constructive intensional type theory of Martin-Löf (MLTT).
- However, there are other approaches in the research programme.
- It is not so easy to say what the essential features of HoTT/UF are.
- Univalence, infinity groupoids, internal language of an $(\infty, 1)$ topos,...

What is HoTT?

- Homotopy Type Theory/Univalent Foundations as presented in the 'HoTT Book' is built on the constructive intensional type theory of Martin-Löf (MLTT).
- However, there are other approaches in the research programme.
- It is not so easy to say what the essential features of HoTT/UF are.
- Univalence, infinity groupoids, internal language of an $(\infty, 1)$ topos,...
- For the purposes of this talk I will consider only the theory of the HoTT book.

Identity in HoTT

- Perhaps the single most conceptually interesting feature of the theory is the way it treats of identity.

- Perhaps the single most conceptually interesting feature of the theory is the way it treats of identity.
- There are two kinds of identity and they do not reflect each other.

- Perhaps the single most conceptually interesting feature of the theory is the way it treats of identity.
- There are two kinds of identity and they do not reflect each other.
- Is identity in HoTT really identity or is it really some other relation such as indiscernibility?

Univalence

- The Univalence Axiom is sometimes said to express that **Isomorphism is Isomorphic to Identity**.

Univalence

- The Univalence Axiom is sometimes said to express that **Isomorphism is Isomorphic to Identity**.
- Steve Awodey has claimed that Univalence captures the essential notion of mathematical structuralism, that "mathematical objects simply *are* structures".

- The Univalence Axiom is sometimes said to express that **Isomorphism is Isomorphic to Identity**.
- Steve Awodey has claimed that Univalence captures the essential notion of mathematical structuralism, that "mathematical objects simply *are* structures".
- However, the above statement of Univalence, expressed in terms of isomorphism, is inconsistent (as Awodey is fully aware). The correct statement of Univalence is instead in terms of *equivalence*, a notion related to isomorphism but subtly different.

Univalence

- The Univalence Axiom is sometimes said to express that **Isomorphism is Isomorphic to Identity**.
- Steve Awodey has claimed that Univalence captures the essential notion of mathematical structuralism, that "mathematical objects simply *are* structures".
- However, the above statement of Univalence, expressed in terms of isomorphism, is inconsistent (as Awodey is fully aware). The correct statement of Univalence is instead in terms of *equivalence*, a notion related to isomorphism but subtly different.
- What is the difference between equivalence and isomorphism, and does mathematical structuralism motivate Univalence?

HoTT

- HoTT is an **intensional constructive dependent type theory** that can be interpreted in homotopy theory – but can also be understood without homotopy.

- HoTT is an **intensional constructive dependent type theory** that can be interpreted in homotopy theory – but can also be understood without homotopy.
- We can roughly think of types as ‘kinds of thing that a mathematical entity could be’.

HoTT

- HoTT is an **intensional constructive dependent type theory** that can be interpreted in homotopy theory – but can also be understood without homotopy.
- We can roughly think of types as 'kinds of thing that a mathematical entity could be'.
- For example, there is a type $\mathbb{N}$, the natural numbers, whose tokens are individual natural numbers e.g. $5 : \mathbb{N}$, $24 : \mathbb{N}$, etc.

HoTT

- HoTT is an **intensional constructive dependent type theory** that can be interpreted in homotopy theory – but can also be understood without homotopy.
- We can roughly think of types as ‘kinds of thing that a mathematical entity could be’.
- For example, there is a type $\mathbb{N}$, the natural numbers, whose tokens are individual natural numbers e.g. $5 : \mathbb{N}$, $24 : \mathbb{N}$, etc.
- Another example: given some $\mathtt{A}$ and $\mathtt{B}$ we can define the type $\mathtt{Iso}(\mathtt{A}, \mathtt{B})$, which is the type of isomorphisms between $\mathtt{A}$ and $\mathtt{B}$.

Propositions

- Rather than being built on top of a separate logic, HoTT incorporates logic into the framework.

Propositions

- Rather than being built on top of a separate logic, HoTT incorporates logic into the framework.
- As well as types like $\mathbb{N}$ there are also types corresponding to **propositions**. The tokens of these types are **certificates** to the truth of the proposition. A proposition may have multiple certificates.

Propositions

- Rather than being built on top of a separate logic, HoTT incorporates logic into the framework.
- As well as types like $\mathbb{N}$ there are also types corresponding to **propositions**. The tokens of these types are **certificates** to the truth of the proposition. A proposition may have multiple certificates.
- Some types can be read in both ways:
  e.g. `Iso(A, B)` is both the proposition '`A` *and* `B` *are isomorphic*' and the type of *isomorphisms between* `A` *and* `B`.

Propositions

- Rather than being built on top of a separate logic, HoTT incorporates logic into the framework.
- As well as types like $\mathbb{N}$ there are also types corresponding to **propositions**. The tokens of these types are **certificates** to the truth of the proposition. A proposition may have multiple certificates.
- Some types can be read in both ways:
  e.g. $\texttt{Iso}(\texttt{A}, \texttt{B})$ is both the proposition '$\texttt{A}$ *and* $\texttt{B}$ *are isomorphic*' and the type of *isomorphisms between* $\texttt{A}$ *and* $\texttt{B}$.
- The rules of type formation and token construction then correspond to the basic operations of logic such as conjunction, disjunction, and implication.

The Language of HoTT

Starting from the **Curry-Howard correspondence**, HoTT has type constructions corresponding to logical connectives, to Existential and Universal quantification, and to equality/identity statements.

Logic	Type theory	Notation
Implication	Function type	$A \to B$
Conjunction	Product	$A \times B$
Disjunction	Coproduct	$A + B$
Existential quantification	Dependent pair type	$\sum_{x:A} P(x)$
Universal quantification	Dependent function type	$\prod_{x:A} P(x)$
Identity	Identity type	$Id_A(x, y)$

The Language of HoTT

Starting from the **Curry-Howard correspondence**, HoTT has type constructions corresponding to logical connectives, to Existential and Universal quantification, and to equality/identity statements.

Logic	Type theory	Notation
Implication	Function type	$\mathtt{A} \to \mathtt{B}$
Conjunction	Product	$\mathtt{A} \times \mathtt{B}$
Disjunction	Coproduct	$\mathtt{A} + \mathtt{B}$
Existential quantification	Dependent pair type	$\sum_{\mathtt{x:A}} \mathtt{P(x)}$
Universal quantification	Dependent function type	$\prod_{\mathtt{x:A}} \mathtt{P(x)}$
Identity	Identity type	$\mathtt{Id_A(x, y)}$

The resulting logic is constructive.

Negation

- If type P corresponds to some proposition then the negation of that proposition is given by the function type $P \to 0$, where 0 is the **Zero type**, which has no tokens.

Negation

- If type $\mathtt{P}$ corresponds to some proposition then the negation of that proposition is given by the function type $\mathtt{P} \to \mathtt{0}$, where $\mathtt{0}$ is the **Zero type**, which has no tokens.

Negation

- If type P corresponds to some proposition then the negation of that proposition is given by the function type $\mathtt{P} \to \mathtt{0}$, where 0 is the **Zero type**, which has no tokens.
- Thus we take it as definitve of negation that it satisfies the Law of Non-Contradiction: if we had both a token of P and a token of $\mathtt{P} \to \mathtt{0}$ then we could, impossibly, produce a token of 0.

Negation

- If type $\mathtt{P}$ corresponds to some proposition then the negation of that proposition is given by the function type $\mathtt{P} \to \mathtt{0}$, where $\mathtt{0}$ is the **Zero type**, which has no tokens.
- Thus we take it as definitve of negation that it satisfies the Law of Non-Contradiction: if we had both a token of $\mathtt{P}$ and a token of $\mathtt{P} \to \mathtt{0}$ then we could, impossibly, produce a token of $\mathtt{0}$.
- The Law of Excluded Middle does not hold in general in HoTT. There may be types $\mathtt{X}$ such that we cannot produce a token of $\mathtt{X}$ nor a token of $\mathtt{X} \to \mathtt{0}$.

Negation

- If type $\mathtt{P}$ corresponds to some proposition then the negation of that proposition is given by the function type $\mathtt{P} \rightarrow \mathtt{0}$, where $\mathtt{0}$ is the **Zero type**, which has no tokens.
- Thus we take it as definitve of negation that it satisfies the Law of Non-Contradiction: if we had both a token of $\mathtt{P}$ and a token of $\mathtt{P} \rightarrow \mathtt{0}$ then we could, impossibly, produce a token of $\mathtt{0}$.
- The Law of Excluded Middle does not hold in general in HoTT. There may be types $\mathtt{X}$ such that we cannot produce a token of $\mathtt{X}$ nor a token of $\mathtt{X} \rightarrow \mathtt{0}$.
- Similarly, given a function of type $(\mathtt{X} \rightarrow \mathtt{0}) \rightarrow \mathtt{0}$ we cannot in general produce a token of $\mathtt{X}$. Thus we do not have Double Negation Elimination.

Negation

- If type P corresponds to some proposition then the negation of that proposition is given by the function type $P \to 0$, where 0 is the **Zero type**, which has no tokens.
- Thus we take it as definitve of negation that it satisfies the Law of Non-Contradiction: if we had both a token of P and a token of $P \to 0$ then we could, impossibly, produce a token of 0.
- The Law of Excluded Middle does not hold in general in HoTT. There may be types X such that we cannot produce a token of X nor a token of $X \to 0$.
- Similarly, given a function of type $(X \to 0) \to 0$ we cannot in general produce a token of X. Thus we do not have Double Negation Elimination.
- This is the sense in which the logic of HoTT is **constructive**.

Constructivity of HoTT

- To prove a disjunction $(\mathtt{A} + \mathtt{B})$ we must be able to prove at least one of the disjuncts.
- A function of type $\mathtt{A} \to \mathtt{B}$ is an algorithm or procedure that, when given a token of $\mathtt{A}$, produces a token of type $\mathtt{B}$.
- While the Law of Excluded Middle and Double Negation Elimination are not laws of the logic of HoTT, we are always free to posit any particular instance of LEM or DNE as a premise: we can use classical principles, we just have to be explicit about these uses.
- Constructivism need not be motivated by any anti-classical or Brouwerian sentiments.

Intensional

- The type theory is *intensional* in the sense that types are individuated by how they are conceived and described.

Intensional

- The type theory is *intensional* in the sense that types are individuated by how they are conceived and described.
- Thus for example two types with the same extension may be considered distinct (e.g. ‘even divisors of 9’ and ‘even divisors of 13’)

Intensional

- The type theory is *intensional* in the sense that types are individuated by how they are conceived and described.
- Thus for example two types with the same extension may be considered distinct (e.g. 'even divisors of 9' and 'even divisors of 13')
- Something having a property (e.g. a number being prime) is expressed in the theory by a pair consisting of the thing and the certificate to the proposition that it has that property.

Intensional

- The type theory is *intensional* in the sense that types are individuated by how they are conceived and described.
- Thus for example two types with the same extension may be considered distinct (e.g. 'even divisors of 9' and 'even divisors of 13')
- Something having a property (e.g. a number being prime) is expressed in the theory by a pair consisting of the thing and the certificate to the proposition that it has that property.
- For example, we have the type of *prime numbers* $\mathbb{P}$ whose tokens are pairs such as $(5, \mathtt{p}_5) : \mathbb{P}$, and $(17, \mathtt{p}_{17}) : \mathbb{P}$, where $\mathtt{p}_\mathtt{n}$ is a certificate to the fact that $\mathtt{n} : \mathbb{N}$ is prime.

Intensional

- The type theory is *intensional* in the sense that types are individuated by how they are conceived and described.
- Thus for example two types with the same extension may be considered distinct (e.g. 'even divisors of 9' and 'even divisors of 13')
- Something having a property (e.g. a number being prime) is expressed in the theory by a pair consisting of the thing and the certificate to the proposition that it has that property.
- For example, we have the type of *prime numbers* $\mathbb{P}$ whose tokens are pairs such as $(5, \mathtt{p}_5) : \mathbb{P}$, and $(17, \mathtt{p}_{17}) : \mathbb{P}$, where $\mathtt{p}_\mathtt{n}$ is a certificate to the fact that $\mathtt{n} : \mathbb{N}$ is prime.
- The theory is **proof-relevant** – it matters which certificate to a proposition we have. e.g. when A and B are isomorphic it matters *which* isomorphism between them has been constructed.

Products and ordered pairs

- Given any two types $\mathtt{A}$ and $\mathtt{B}$ we can form the **product** type $\mathtt{A} \times \mathtt{B}$. Tokens of this are ordered pairs $(\mathtt{a}, \mathtt{b}) : \mathtt{A} \times \mathtt{B}$, where $\mathtt{a} : \mathtt{A}$ and $\mathtt{b} : \mathtt{B}$.

- Given any two types $\mathtt{A}$ and $\mathtt{B}$ we can form the **product** type $\mathtt{A} \times \mathtt{B}$. Tokens of this are ordered pairs $(\mathtt{a}, \mathtt{b}) : \mathtt{A} \times \mathtt{B}$, where $\mathtt{a} : \mathtt{A}$ and $\mathtt{b} : \mathtt{B}$.
- This is one of the basic constructions in HoTT, not derivative as in ZFC set theory (where ordered pairs are sets of a certain form).
- Note that the product $\mathtt{A} \times \mathtt{B}$ is distinct from the product $\mathtt{B} \times \mathtt{A}$ so philosophers would call the theory ‘hyperintensional’. However, these products are equivalent and under univalence they are identified.

Internal and External Identity

- HoTT has two different identity relations:

Internal and External Identity

- HoTT has two different identity relations:
    - **external identity** $\mathtt{x} \equiv \mathtt{y}$ says '$\mathtt{x}$' and '$\mathtt{y}$' are two expressions naming the same thing

Internal and External Identity

- HoTT has two different identity relations:
  - **external identity** $\mathtt{x} \equiv \mathtt{y}$ says '$\mathtt{x}$' and '$\mathtt{y}$' are two expressions naming the same thing
  - **internal identity** $\mathtt{x} = \mathtt{y}$ is a proposition represented by a type $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$ in the theory

Internal and External Identity

- HoTT has two different identity relations:
  - **external identity** $\mathtt{x} \equiv \mathtt{y}$ says '$\mathtt{x}$' and '$\mathtt{y}$' are two expressions naming the same thing
  - **internal identity** $\mathtt{x} = \mathtt{y}$ is a proposition represented by a type $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$ in the theory
- Externally identical elements may be substituted for one another in any context, with no restrictions.

Internal and External Identity

- HoTT has two different identity relations:
  - **external identity** $x \equiv y$ says 'x' and 'y' are two expressions naming the same thing
  - **internal identity** $x = y$ is a proposition represented by a type $\mathtt{Id}_{\mathtt{A}}(x, y)$ in the theory
- Externally identical elements may be substituted for one another in any context, with no restrictions.
- Internal identity is not so simple, but this is from where much of the interest and power of HoTT comes.

Internal and External Identity

- HoTT has two different identity relations:
  - **external identity** $\mathtt{x} \equiv \mathtt{y}$ says '$\mathtt{x}$' and '$\mathtt{y}$' are two expressions naming the same thing
  - **internal identity** $\mathtt{x} = \mathtt{y}$ is a proposition represented by a type $\mathtt{Id_A}(\mathtt{x}, \mathtt{y})$ in the theory
- Externally identical elements may be substituted for one another in any context, with no restrictions.
- Internal identity is not so simple, but this is from where much of the interest and power of HoTT comes.
- Importantly, internal identity does not imply external identity. This is another sense in which the theory is *intensional*. Thus in HoTT internally identifying two tokens of a type does not amount to just collapsing them together.

Identity Types

- For any tokens $\mathtt{x}, \mathtt{y}$ of any type $\mathtt{A}$ we can form the proposition that they are identical. The type corresponding to this proposition is the *identity type* $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$.

Identity Types

- For any tokens $\mathtt{x}, \mathtt{y}$ of any type $\mathtt{A}$ we can form the proposition that they are identical. The type corresponding to this proposition is the *identity type* $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$.
- A token of an identity type is a certificate to that proposition (an 'identification'). For example, for any token $\mathtt{x} : \mathtt{A}$ we have the *trivial self-identification* $\mathtt{refl}_{\mathtt{x}} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{x})$.

Identity Types

- For any tokens $\mathtt{x}, \mathtt{y}$ of any type $\mathtt{A}$ we can form the proposition that they are identical. The type corresponding to this proposition is the *identity type* $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$.
- A token of an identity type is a certificate to that proposition (an 'identification'). For example, for any token $\mathtt{x} : \mathtt{A}$ we have the *trivial self-identification* $\mathtt{refl}_{\mathtt{x}} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{x})$.
- For two tokens that are distinct the corresponding identity type will be uninhabited, but two identical tokens may have *multiple* identifications.

Identity Types

- For any tokens $\mathtt{x}, \mathtt{y}$ of any type $\mathtt{A}$ we can form the proposition that they are identical. The type corresponding to this proposition is the *identity type* $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$.
- A token of an identity type is a certificate to that proposition (an 'identification'). For example, for any token $\mathtt{x} : \mathtt{A}$ we have the *trivial self-identification* $\mathtt{refl}_{\mathtt{x}} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{x})$.
- For two tokens that are distinct the corresponding identity type will be uninhabited, but two identical tokens may have *multiple* identifications.
- Every token of every type is trivially identical to itself, but *non-trivial* identifications are also allowed.

Identity Types

- For any tokens $\mathtt{x}, \mathtt{y}$ of any type $\mathtt{A}$ we can form the proposition that they are identical. The type corresponding to this proposition is the *identity type* $\mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$.
- A token of an identity type is a certificate to that proposition (an 'identification'). For example, for any token $\mathtt{x} : \mathtt{A}$ we have the *trivial self-identification* $\mathtt{refl}_{\mathtt{x}} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{x})$.
- For two tokens that are distinct the corresponding identity type will be uninhabited, but two identical tokens may have *multiple* identifications.
- Every token of every type is trivially identical to itself, but *non-trivial* identifications are also allowed.
- Nothing we can express within the language of HoTT can distinguish tokens that are identified. In particular, all functions respect identity (i.e. $\mathtt{x} = \mathtt{y}$ implies $\mathtt{f}(\mathtt{x}) = \mathtt{f}(\mathtt{y})$).

Higher Identity types

- We can form the identity type for any tokens of any type. Since identifications are themselves tokens of a type we can form identities between them as well.

Higher Identity types

- We can form the identity type for any tokens of any type. Since identifications are themselves tokens of a type we can form identities between them as well.
- If we have identifications $\mathtt{p}, \mathtt{q} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$ we can form the *higher identity type*

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})}(\mathtt{p}, \mathtt{q})$$

Higher Identity types

- We can form the identity type for any tokens of any type. Since identifications are themselves tokens of a type we can form identities between them as well.
- If we have identifications $\mathtt{p}, \mathtt{q} : \mathtt{Id}_{\mathtt{A}}(\mathtt{x}, \mathtt{y})$ we can form the *higher identity type*

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})}(\mathtt{p}, \mathtt{q})$$

- If we have two tokens α, β of this type then we can form the higher identity type

$$\mathtt{Id}_{\mathtt{Id}_{\mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})}(\mathtt{p},\mathtt{q})}(\alpha, \beta)$$

and so on . . .

Path induction

- To prove theorems involving identifications we use the powerful technique of *path induction*.

Path induction

- To prove theorems involving identifications we use the powerful technique of *path induction*.
- Fix a type A and consider all identifications between tokens of A. If we want to prove that some property holds of all such identifications, path induction says it is sufficient to prove that the property holds of the trivial self-identifications $\mathtt{refl}_{\mathtt{x}}$ for each $\mathtt{x} : \mathtt{A}$.

Path induction

- To prove theorems involving identifications we use the powerful technique of *path induction*.
- Fix a type $\mathtt{A}$ and consider all identifications between tokens of $\mathtt{A}$. If we want to prove that some property holds of all such identifications, path induction says it is sufficient to prove that the property holds of the trivial self-identifications $\mathtt{refl}_{\mathtt{x}}$ for each $\mathtt{x} : \mathtt{A}$.

$$\mathtt{BPI} : \prod_{\mathtt{A}:\mathtt{TYPE}} \prod_{\mathtt{a}:\mathtt{A}} \prod_{\mathtt{K}:\mathtt{E}_\mathtt{a}\to\mathtt{TYPE}} \left[\mathtt{K}(\mathtt{a}, \mathtt{refl}_\mathtt{a}) \to \prod_{(\mathtt{b},\mathtt{p}):\mathtt{E}_\mathtt{a}} \mathtt{K}(\mathtt{b}, \mathtt{p}) \right]$$

Path induction

- To prove theorems involving identifications we use the powerful technique of *path induction*.
- Fix a type $\mathtt{A}$ and consider all identifications between tokens of $\mathtt{A}$. If we want to prove that some property holds of all such identifications, path induction says it is sufficient to prove that the property holds of the trivial self-identifications $\mathtt{refl_x}$ for each $\mathtt{x} : \mathtt{A}$.

$$\mathtt{BPI} : \prod_{\mathtt{A:TYPE}} \prod_{\mathtt{a:A}} \prod_{\mathtt{K:E_a \to TYPE}} \left[\mathtt{K}(\mathtt{a}, \mathtt{refl_a}) \to \prod_{(\mathtt{b,p}):\mathtt{E_a}} \mathtt{K}(\mathtt{b}, \mathtt{p}) \right]$$

- This often makes such proofs very simple or even trivial. The justification of path induction in the official presentation of HoTT appeals to the homotopy interpretation but it can be otherwise justified.

Path Induction and Intensionality

- It may seem that path induction says that all identifications are trivial, but this doesn't follow.

Path Induction and Intensionality

- It may seem that path induction says that all identifications are trivial, but this doesn't follow.
- To prove this we would need a predicate $\mathtt{Q}$ that says of any identification of some token $\mathtt{b}$ of $\mathtt{A}$ with $\mathtt{a}$, $\mathtt{p} : \mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ that it's identical to the trivial self-identification of $\mathtt{a}$.

Path Induction and Intensionality

- It may seem that path induction says that all identifications are trivial, but this doesn't follow.
- To prove this we would need a predicate $\mathtt{Q}$ that says of any identification of some token $\mathtt{b}$ of $\mathtt{A}$ with $\mathtt{a}$, $\mathtt{p} : \mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ that it's identical to the trivial self-identification of $\mathtt{a}$.

$$\mathtt{Q}(\mathtt{a}, \mathtt{b}, \mathtt{p}) :\equiv \mathtt{Id}_{\mathtt{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{a})}(\mathtt{p}, \mathtt{refl}_{\mathtt{a}})$$

Path Induction and Intensionality

- It may seem that path induction says that all identifications are trivial, but this doesn't follow.
- To prove this we would need a predicate $\mathtt{Q}$ that says of any identification of some token $\mathtt{b}$ of $\mathtt{A}$ with $\mathtt{a}$, $\mathtt{p} : \mathtt{Id_A}(\mathtt{a}, \mathtt{b})$ that it's identical to the trivial self-identification of $\mathtt{a}$.

$$\mathtt{Q}(\mathtt{a}, \mathtt{b}, \mathtt{p}) :\equiv \mathtt{Id}_{\mathtt{Id_A}(\mathtt{a},\mathtt{a})}(\mathtt{p}, \mathtt{refl_a})$$

- But such a predicate is not well-typed, because $\mathtt{p}$ is not a token of $\mathtt{Id_A}(\mathtt{a}, \mathtt{a})$.

Path Induction and Intensionality

- It may seem that path induction says that all identifications are trivial, but this doesn't follow.
- To prove this we would need a predicate $\mathtt{Q}$ that says of any identification of some token $\mathtt{b}$ of $\mathtt{A}$ with $\mathtt{a}$, $\mathtt{p} : \mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ that it's identical to the trivial self-identification of $\mathtt{a}$.

$$\mathtt{Q}(\mathtt{a}, \mathtt{b}, \mathtt{p}) :\equiv \mathtt{Id}_{\mathtt{Id}_{\mathtt{A}}(\mathtt{a},\mathtt{a})}(\mathtt{p}, \mathtt{refl}_{\mathtt{a}})$$

- But such a predicate is not well-typed, because $\mathtt{p}$ is not a token of $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{a})$.
- However, in an *extensional* type theory, from $\mathtt{p} : \mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ we could derive $\mathtt{a} \equiv \mathtt{b}$. Then the above $\mathtt{Q}$ would be well-typed, and the proof that all identifications are trivial goes through.

Is 'Identity' Really Identity?

The internal identity relation in HoTT represented by the identity types $\mathtt{Id_A(a, b)}$ behaves radically differently from our intuitive view of identity:

- There are *two* notions of identity in HoTT – internal and external.

Is 'Identity' Really Identity?

The internal identity relation in HoTT represented by the identity types $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ behaves radically differently from our intuitive view of identity:

- There are *two* notions of identity in HoTT – internal and external.
- Whereas conventionally identity is a binary yes/no property, in HoTT two tokens may have *multiple identifications*, some of which themselves may be identified, and so on.

Is ‘Identity’ Really Identity?

The internal identity relation in HoTT represented by the identity types $\mathtt{Id_A}(\mathtt{a}, \mathtt{b})$ behaves radically differently from our intuitive view of identity:

- There are *two* notions of identity in HoTT – internal and external.
- Whereas conventionally identity is a binary yes/no property, in HoTT two tokens may have *multiple identifications*, some of which themselves may be identified, and so on.
- While internal identity is reflexive (and thus every token has a trivial self-identification) tokens may also have *non-trivial self-identifications*.

Is 'Identity' Really Identity?

The internal identity relation in HoTT represented by the identity types $\mathtt{Id_A(a, b)}$ behaves radically differently from our intuitive view of identity:

- There are *two* notions of identity in HoTT – internal and external.
- Whereas conventionally identity is a binary yes/no property, in HoTT two tokens may have *multiple identifications*, some of which themselves may be identified, and so on.
- While internal identity is reflexive (and thus every token has a trivial self-identification) tokens may also have *non-trivial self-identifications*.
- The *Univalence Axiom* (which we'll discuss later) says that two types may be identified in virtue of being isomorphic.

Is 'Identity' Really Identity?

- While these facts may be given mathematical justification and thus be rendered quite acceptable to the mathematician, they are nonetheless philosophically very puzzling and call into question the name 'identity' for the relation.

Is 'Identity' Really Identity?

- While these facts may be given mathematical justification and thus be rendered quite acceptable to the mathematician, they are nonetheless philosophically very puzzling and call into question the name 'identity' for the relation.
- If external identity is the relation that says that two expressions 'name the same token or type' and can be freely inter-substituted, how can some other relation also be 'identity'? And how can two entities be distinct with respect to the former while 'identical' with respect to the latter?

Is 'Identity' Really Identity?

- While these facts may be given mathematical justification and thus be rendered quite acceptable to the mathematician, they are nonetheless philosophically very puzzling and call into question the name 'identity' for the relation.
- If external identity is the relation that says that two expressions 'name the same token or type' and can be freely inter-substituted, how can some other relation also be 'identity'? And how can two entities be distinct with respect to the former while 'identical' with respect to the latter?
- Perhaps these unusual features indicate that the relation represented by $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ is not identity at all but rather *indiscernibility*.

Is 'Identity' Really Identity?

- While these facts may be given mathematical justification and thus be rendered quite acceptable to the mathematician, they are nonetheless philosophically very puzzling and call into question the name 'identity' for the relation.
- If external identity is the relation that says that two expressions 'name the same token or type' and can be freely inter-substituted, how can some other relation also be 'identity'? And how can two entities be distinct with respect to the former while 'identical' with respect to the latter?
- Perhaps these unusual features indicate that the relation represented by $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ is not identity at all but rather *indiscernibility*.
- If we read $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$ as indiscernibility instead of identity then the above features are no longer unexpected, and fits better with standard views.

Identity as Indiscernibility?

- If internal identity is understood as indiscernibility then external identity becomes the single ‘true’ identity relation in the theory, and it is at least conceivable that two non-identical tokens might be indiscernible.

Identity as Indiscernibility?

- If internal identity is understood as indiscernibility then external identity becomes the single 'true' identity relation in the theory, and it is at least conceivable that two non-identical tokens might be indiscernible.
- The fact that two tokens are indiscernible may be witnessed in multiple different ways – for example, by different isomorphisms between them – and it is quite natural to study the correspondences between distinct isomorphisms.

Identity as Indiscernibility?

- If internal identity is understood as indiscernibility then external identity becomes the single 'true' identity relation in the theory, and it is at least conceivable that two non-identical tokens might be indiscernible.
- The fact that two tokens are indiscernible may be witnessed in multiple different ways – for example, by different isomorphisms between them – and it is quite natural to study the correspondences between distinct isomorphisms.
- While any entity is trivially indiscernible from itself, if it has any non-trivial automorphisms then these provide additional ways in which the self-indiscernibility may be witnessed.

Identity as Indiscernibility?

- If internal identity is understood as indiscernibility then external identity becomes the single ‘true’ identity relation in the theory, and it is at least conceivable that two non-identical tokens might be indiscernible.
- The fact that two tokens are indiscernible may be witnessed in multiple different ways – for example, by different isomorphisms between them – and it is quite natural to study the correspondences between distinct isomorphisms.
- While any entity is trivially indiscernible from itself, if it has any non-trivial automorphisms then these provide additional ways in which the self-indiscernibility may be witnessed.
- On this view of $\mathtt{Id}_{\mathtt{A}}(\mathtt{a}, \mathtt{b})$, the Univalence Axiom says that types that are equivalent to each other (in a certain sense) are indiscernible from each other

Minimalism about Identity

- This view of Identity-as-Indiscernibility may be understood as saying that "facts about individuation are exhausted by facts about (non-identity-involving) properties".

Minimalism about Identity

- This view of Identity-as-Indiscernibility may be understood as saying that "facts about individuation are exhausted by facts about (non-identity-involving) properties".
- In other words, if two entities have all their (non-identity-involving) properties in common, then they should be counted as 'identical'.

Minimalism about Identity

- This view of Identity-as-Indiscernibility may be understood as saying that "facts about individuation are exhausted by facts about (non-identity-involving) properties".
- In other words, if two entities have all their (non-identity-involving) properties in common, then they should be counted as 'identical'.
- To study this further we should formalise the definition of *indiscernibility* and the Principle of the Identity of Indiscernibles (PII).

Discernibility and Indiscernibility

- The natural definition of discernibility to use is *absolute discernibility*, which we formalise as:

$$\mathtt{Dis_A(a, b)} :\equiv \sum_{\mathtt{P:A}\to\mathcal{U}} \mathtt{P(a)} \times \neg\mathtt{P(b)}$$

"there is some property $\mathtt{P}$ that holds of $\mathtt{a}$ and not of $\mathtt{b}$".

Discernibility and Indiscernibility

- The natural definition of discernibility to use is *absolute discernibility*, which we formalise as:

$$\mathtt{Dis_A(a, b)} :\equiv \sum_{\mathtt{P:A}\to\mathcal{U}} \mathtt{P(a)} \times \neg\mathtt{P(b)}$$

  "there is some property $\mathtt{P}$ that holds of $\mathtt{a}$ and not of $\mathtt{b}$".

- However, since the logic of HoTT is constructive, we should not define indiscernibility as simply the negation of discernibility, as this will in general be too weak.

Discernibility and Indiscernibility

- The natural definition of discernibility to use is *absolute discernibility*, which we formalise as:

$$\mathtt{Dis}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}) :\equiv \sum_{\mathtt{P}:\mathtt{A}\to\mathcal{U}} \mathtt{P}(\mathtt{a}) \times \neg\mathtt{P}(\mathtt{b})$$

"there is some property P that holds of a and not of b".

- However, since the logic of HoTT is constructive, we should not define indiscernibility as simply the negation of discernibility, as this will in general be too weak.
- Rather, we define indiscernibility as

$$\mathtt{InDis}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}) :\equiv \prod_{\mathtt{P}:\mathtt{A}\to\mathcal{U}} \mathtt{P}(\mathtt{a}) \leftrightarrow \mathtt{P}(\mathtt{b})$$

"for every property P, token a satisfies the property iff b does".

PII: Principle of Identity of Indiscernibles

- The original formulation of PII by Leibniz is:

  > *"it is not true that two substances may be exactly alike and differ only numerically, solo numero"*

  or in other words there are no two (non-identical) objects that share all their properties.

PII: Principle of Identity of Indiscernibles

- The original formulation of PII by Leibniz is:

  > *"it is not true that two substances may be exactly alike and differ only numerically, solo numero"*

  or in other words there are no two (non-identical) objects that share all their properties.

- We can formalise this as:

$$\neg \sum_{\mathtt{x},\mathtt{y}:\mathtt{A}} \big(\mathtt{InDis}_{\mathtt{A}}(\mathtt{x},\mathtt{y}) \times \neg \mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})\big)$$

PII: Principle of Identity of Indiscernibles

- The original formulation of PII by Leibniz is:

  > *"it is not true that two substances may be exactly alike and differ only numerically, solo numero"*

  or in other words there are no two (non-identical) objects that share all their properties.

- We can formalise this as:

$$\neg \sum_{\mathtt{x},\mathtt{y}:\mathtt{A}} \left(\mathtt{InDis}_{\mathtt{A}}(\mathtt{x},\mathtt{y}) \times \neg \mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})\right)$$

- Classically this is equivalent to "if any two objects are indiscernible then they are identical":

$$\prod_{\mathtt{x},\mathtt{y}:\mathtt{A}} \left(\mathtt{InDis}_{\mathtt{A}}(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})\right)$$

PII: Principle of Identity of Indiscernibles

- The original formulation of PII by Leibniz is:

  > *"it is not true that two substances may be exactly alike and differ only numerically, solo numero"*

  or in other words there are no two (non-identical) objects that share all their properties.

- We can formalise this as:

$$\neg \sum_{\mathtt{x},\mathtt{y}:\mathtt{A}} \left(\mathtt{InDis}_{\mathtt{A}}(\mathtt{x},\mathtt{y}) \times \neg \mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})\right)$$

- Classically this is equivalent to "if any two objects are indiscernible then they are identical":

$$\prod_{\mathtt{x},\mathtt{y}:\mathtt{A}} \left(\mathtt{InDis}_{\mathtt{A}}(\mathtt{x},\mathtt{y}) \to \mathtt{Id}_{\mathtt{A}}(\mathtt{x},\mathtt{y})\right)$$

- Constructively the latter statement is *stronger* than the former.

Problems with Identity-as-Indiscernibility

- The behaviour of $\mathtt{InDis_A}(\mathtt{a}, \mathtt{b})$ is importantly different from that of $\mathtt{Id_A}(\mathtt{a}, \mathtt{b})$; specifically, it does not support *path induction*.

Problems with Identity-as-Indiscernibility

- The behaviour of $\texttt{InDis}_{\texttt{A}}(\texttt{a}, \texttt{b})$ is importantly different from that of $\texttt{Id}_{\texttt{A}}(\texttt{a}, \texttt{b})$; specifically, it does not support *path induction*.
- The role of *universes* in the definition of indiscernibility.

Indexing to Universes

- There is a substantial difference between identity and indiscernibility that obstructs their unification.

Indexing to Universes

- There is a substantial difference between identity and indiscernibility that obstructs their unification.
- The definition of `InDis`,

$$\mathtt{InDis}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}) :\equiv \prod_{\mathtt{P}:\mathtt{A}\to\mathcal{U}} \mathtt{P}(\mathtt{a}) \leftrightarrow \mathtt{P}(\mathtt{b})$$

involves a quantification over all predicates defined on type $\mathtt{A}$.

Indexing to Universes

- There is a substantial difference between identity and indiscernibility that obstructs their unification.
- The definition of $\mathtt{InDis}$,

$$\mathtt{InDis}_{\mathtt{A}}(\mathtt{a}, \mathtt{b}) :\equiv \prod_{\mathtt{P}:\mathtt{A} \to \mathcal{U}} \mathtt{P}(\mathtt{a}) \leftrightarrow \mathtt{P}(\mathtt{b})$$

  involves a quantification over all predicates defined on type $\mathtt{A}$.
- The type of the predicates that we quantify over is written as $\mathtt{A} \to \mathcal{U}$, where $\mathcal{U}$ is the *universe* – roughly, the type whose tokens are all the types under consideration.

Identity and the classification of types

- Types can be classified according to how rich their identity structure is.

Identity and the classification of types

- Types can be classified according to how rich their identity structure is.
- There are singletons, mere propositions, sets, groupoids and so on.

Identity and the classification of types

- Types can be classified according to how rich their identity structure is.
- There are singletons, mere propositions, sets, groupoids and so on.
- If identity was replaced by indiscernibility then these definitions would be relative to universes.

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:
  1. failure to recover path induction (which is an essential feature of how identity types are used in HoTT)

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:
  1. failure to recover path induction (which is an essential feature of how identity types are used in HoTT)
  2. indiscernibility is only relative to some universe whereas identity is not relativised in this way.

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:
  1. failure to recover path induction (which is an essential feature of how identity types are used in HoTT)
  2. indiscernibility is only relative to some universe whereas identity is not relativised in this way.

(More details in our paper *Identity in HoTT part 2: The Conceptual and Philosophical Status of Identity in HoTT*, in *Philosophia Mathematica*.)

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:
  1. failure to recover path induction (which is an essential feature of how identity types are used in HoTT)
  2. indiscernibility is only relative to some universe whereas identity is not relativised in this way.

  (More details in our paper *Identity in HoTT part 2: The Conceptual and Philosophical Status of Identity in HoTT*, in *Philosophia Mathematica*.)

- Thus we cannot resolve the counterintuitive features of 'identity types' in HoTT by taking them to stand instead for indiscernibility.

Identity in HoTT is not Indiscernibility

- Two important differences between identity and indiscernibility:
  1. failure to recover path induction (which is an essential feature of how identity types are used in HoTT)
  2. indiscernibility is only relative to some universe whereas identity is not relativised in this way.

  (More details in our paper *Identity in HoTT part 2: The Conceptual and Philosophical Status of Identity in HoTT*, in *Philosophia Mathematica*.)
- Thus we cannot resolve the counterintuitive features of 'identity types' in HoTT by taking them to stand instead for indiscernibility.
- If we are to take HoTT seriously as a candidate foundation for mathematics then we must get used to the novel way it treats identity instead of reinterpreting away these unusual features.

Non-trivial identities are a promising way to think about symmetry in mathematics and physics.

Non-trivial identities are a promising way to think about symmetry in mathematics and physics.

They may also be applicable to Frege's puzzle.

Non-trivial identities are a promising way to think about symmetry in mathematics and physics.

They may also be applicable to Frege's puzzle.

HoTT as a Foundation for Mathematics?

- HoTT has been proposed as a new foundation for mathematics: the HoTT Book is subtitled "Univalent Foundations of Mathematics".

HoTT as a Foundation for Mathematics?

- HoTT has been proposed as a new foundation for mathematics: the HoTT Book is subtitled "Univalent Foundations of Mathematics".
- Awodey and Voevodsky have given several considerations to motivate the need for HoTT as a new foundation.

- Most working mathematicians don't work in ZFC or category theory or any other official foundational system. They work informally, and if they appeal to the existence of these foundational systems at all it is as an 'in principle' guarantee that their work is secure.

Mismatch between Foundation & Practice

- Most working mathematicians don't work in ZFC or category theory or any other official foundational system. They work informally, and if they appeal to the existence of these foundational systems at all it is as an 'in principle' guarantee that their work is secure.
- There can be a very large gap between what's done in practice and the 'in principle' formal underpinnings.

Abuse of Notation and other Sloppiness

- Working informally may involve (what some might consider) sloppiness, such as identifying isomorphic or equivalent things by 'abuse of notation'.
- Having proved that two structures are isomorphic we often use them interchangeably, ignoring the distinction between them and treating them as identical.

Abuse of Notation and other Sloppiness

- Working informally may involve (what some might consider) sloppiness, such as identifying isomorphic or equivalent things by 'abuse of notation'.
- Having proved that two structures are isomorphic we often use them interchangeably, ignoring the distinction between them and treating them as identical.
- Sometimes we do this by a kind of 'controlled sloppiness', using the same symbol for both structures and simply remembering (in the back of our mind) that there are actually two distinct entities.

Abuse of Notation and other Sloppiness

- Working informally may involve (what some might consider) sloppiness, such as identifying isomorphic or equivalent things by 'abuse of notation'.
- Having proved that two structures are isomorphic we often use them interchangeably, ignoring the distinction between them and treating them as identical.
- Sometimes we do this by a kind of 'controlled sloppiness', using the same symbol for both structures and simply remembering (in the back of our mind) that there are actually two distinct entities.
- For example, Dedekind cuts are sets of rationals, whereas Cauchy sequences are sequences of rationals. When we talk about 'the reals' we freely switch back and forth between these two structures.

Computer checking

- This is a problem if we want computers to help verify our mathematical work.

Computer checking

- This is a problem if we want computers to help verify our mathematical work.
- Much of mathematics is so complex and so specialised that the only reliable defence against error is formal proof checking.

Computer checking

- This is a problem if we want computers to help verify our mathematical work.
- Much of mathematics is so complex and so specialised that the only reliable defence against error is formal proof checking.
- Many proof assistants exist but they're difficult to use because of the gap between formalisation and practice.

Computer checking

- This is a problem if we want computers to help verify our mathematical work.
- Much of mathematics is so complex and so specialised that the only reliable defence against error is formal proof checking.
- Many proof assistants exist but they're difficult to use because of the gap between formalisation and practice.
- In particular, they fail to judge two distinct representations of a given structure (e.g. two different definitions of *ordered pair*, or $\mathbb{R}$) as equivalent – they compare the representations, not the thing represented.

Computer checking

- This is a problem if we want computers to help verify our mathematical work.
- Much of mathematics is so complex and so specialised that the only reliable defence against error is formal proof checking.
- Many proof assistants exist but they're difficult to use because of the gap between formalisation and practice.
- In particular, they fail to judge two distinct representations of a given structure (e.g. two different definitions of *ordered pair*, or $\mathbb{R}$) as equivalent – they compare the representations, not the thing represented.
- We need a formal language that relates better to mathematical practice. This is what HoTT promises (thanks in part to the magic of the Univalence Axiom, about which more later.)

Application to Physics

It has also been claimed by Mike Shulman and others that HoTT is a better foundation for mathematics because it helps us avoid conceptual problems in physics such as the hole argument.

What is a Foundation for Mathematics?

- Advocates of different foundational programmes often have different ideas about what it is to provide a foundation, and what is taken to be required is not always made explicit.

What is a Foundation for Mathematics?

- Advocates of different foundational programmes often have different ideas about what it is to provide a foundation, and what is taken to be required is not always made explicit.
- The most minimal idea of foundation is that of a unifying language and conceptual framework.

What is a Foundation for Mathematics?

- Advocates of different foundational programmes often have different ideas about what it is to provide a foundation, and what is taken to be required is not always made explicit.
- The most minimal idea of foundation is that of a unifying language and conceptual framework.
- Let's suppose that an affirmative answer to the question of whether HoTT is adequate as such a framework for mathematics. (See in particular the reconstructions of the natural numbers (Section 1.9), real numbers (Chapter 11), category theory (Chapter 9) and a model of ZFC set theory (Chapter 10).

What is a Foundation for Mathematics?

- Advocates of different foundational programmes often have different ideas about what it is to provide a foundation, and what is taken to be required is not always made explicit.
- The most minimal idea of foundation is that of a unifying language and conceptual framework.
- Let's suppose that an affirmative answer to the question of whether HoTT is adequate as such a framework for mathematics. (See in particular the reconstructions of the natural numbers (Section 1.9), real numbers (Chapter 11), category theory (Chapter 9) and a model of ZFC set theory (Chapter 10).
- Some mathematicians and philosophers think of a foundation as also answering semantic, metaphysical, epistemological, and/or methodological questions about mathematics.
- An important criterion for some is that a foundation be *autonomous*.

Five Components of a Foundation for Mathematics

There are five interrelated components to a foundation for mathematics, and each generates a series of questions that a given putative foundation for mathematics might be expected to answer.

The Framework

A single **framework** in which to cast, and perhaps also develop, some or all of existing mathematics. This framework involves a mathematical language and theory that may be studied in its own right as well.

The Framework

A single **framework** in which to cast, and perhaps also develop, some or all of existing mathematics. This framework involves a mathematical language and theory that may be studied in its own right as well.

Questions:

What is the theory and the language?

The Framework

A single **framework** in which to cast, and perhaps also develop, some or all of existing mathematics. This framework involves a mathematical language and theory that may be studied in its own right as well.

Questions:

What is the theory and the language?

How are complex higher-level elements constructed or composed out of more basic ones?

The Framework

A single **framework** in which to cast, and perhaps also develop, some or all of existing mathematics. This framework involves a mathematical language and theory that may be studied in its own right as well.

Questions:

What is the theory and the language?

How are complex higher-level elements constructed or composed out of more basic ones?

Is it built on a formal logic or not?

The Framework

A single **framework** in which to cast, and perhaps also develop, some or all of existing mathematics. This framework involves a mathematical language and theory that may be studied in its own right as well.

Questions:

What is the theory and the language?

How are complex higher-level elements constructed or composed out of more basic ones?

Is it built on a formal logic or not?

What role, if any, do axioms play?

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:What are the basic concepts and how are they related?

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:What are the basic concepts and how are they related?

Are statements expressed in the language to be understood as potentially having truth values, and if so is the logic bivalent?

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:What are the basic concepts and how are they related?

Are statements expressed in the language to be understood as potentially having truth values, and if so is the logic bivalent?

How are the rules governing the terms to be understood?

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:What are the basic concepts and how are they related?

Are statements expressed in the language to be understood as potentially having truth values, and if so is the logic bivalent?

How are the rules governing the terms to be understood?

What are the identity criteria for the elements of the theory?

Semantics

A semantics in the sense of an account of the basic concepts of the framework, how the theoretical terms of (1) are to be understood, and an account of how the rules that are used to manipulate the concepts are to be understood.

For example, higher level concepts being composed of lower level ones, all concepts being defined in terms of some basic concepts, and so on.

Questions:What are the basic concepts and how are they related?

Are statements expressed in the language to be understood as potentially having truth values, and if so is the logic bivalent?

How are the rules governing the terms to be understood?

What are the identity criteria for the elements of the theory?

In particular, is the theory extensional or intensional?

Metaphysics

A metaphysics that spells out the ontological status of any entities posited in (2).

Metaphysics

A metaphysics that spells out the ontological status of any entities posited in (2).

Questions: Does the metaphysics posit any objects at all? Is the ontology (if any) to be understood as mind-dependent or mind-independent? What is the relationship between mathematical reality (if any) and physical reality?

Epistemology

An epistemology in the sense of an account of how we are able to know the truths (if any) of mathematics, given (2) and (3) (which may also include an account of the applicability of mathematics) and a justification of the axioms and rules of the framework.

Epistemology

An epistemology in the sense of an account of how we are able to know the truths (if any) of mathematics, given (2) and (3) (which may also include an account of the applicability of mathematics) and a justification of the axioms and rules of the framework.

Questions: Given the answers to the above questions, what account, if any, do we give of mathematical knowledge? In particular, if basic entities are posited, how do we know about them?

Epistemology

An epistemology in the sense of an account of how we are able to know the truths (if any) of mathematics, given (2) and (3) (which may also include an account of the applicability of mathematics) and a justification of the axioms and rules of the framework.

Questions: Given the answers to the above questions, what account, if any, do we give of mathematical knowledge? In particular, if basic entities are posited, how do we know about them?

Given the role of proof in mathematical practice, what is the relationship between mathematical knowledge and proof? How are the rules justified given their interpretation (if any)? If there are axioms, what is their epistemological status – for example, are they taken to be known, or are they taken to be merely hypothetical statements that form the antecedent of conditionals?

Epistemology

An epistemology in the sense of an account of how we are able to know the truths (if any) of mathematics, given (2) and (3) (which may also include an account of the applicability of mathematics) and a justification of the axioms and rules of the framework.

Questions: Given the answers to the above questions, what account, if any, do we give of mathematical knowledge? In particular, if basic entities are posited, how do we know about them?

Given the role of proof in mathematical practice, what is the relationship between mathematical knowledge and proof? How are the rules justified given their interpretation (if any)? If there are axioms, what is their epistemological status – for example, are they taken to be known, or are they taken to be merely hypothetical statements that form the antecedent of conditionals?

What is the relationship between mathematical knowledge and knowledge of physical reality?

Methodology

A methodology for mathematical practice based on some or all of the above.

Methodology

A methodology for mathematical practice based on some or all of the above.

Questions: How is the foundation to be used in practice? In particular, how is it to be applied in the physical sciences?

Autonomy

- If a system is supposed to be a foundation then it can't rest upon something else. The foundation should be the terminus of any sequence of 'what' questions and 'why' questions (Mayberry).

Autonomy

- If a system is supposed to be a foundation then it can't rest upon something else. The foundation should be the terminus of any sequence of 'what' questions and 'why' questions (Mayberry).
- Linnebo & Pettigrew consider three kinds of *autonomy* a foundational system may have.

Autonomy

- If a system is supposed to be a foundation then it can't rest upon something else. The foundation should be the terminus of any sequence of 'what' questions and 'why' questions (Mayberry).
- Linnebo & Pettigrew consider three kinds of *autonomy* a foundational system may have.
- Briefly, a presentation of a system is autonomous iff all of its definitions, justification, and interpretation can be given *without appeal to existing mathematics*. A foundation is autonomous iff it can be given an autonomous presentation.

Autonomy

- If a system is supposed to be a foundation then it can't rest upon something else. The foundation should be the terminus of any sequence of 'what' questions and 'why' questions (Mayberry).
- Linnebo & Pettigrew consider three kinds of *autonomy* a foundational system may have.
- Briefly, a presentation of a system is autonomous iff all of its definitions, justification, and interpretation can be given *without appeal to existing mathematics*. A foundation is autonomous iff it can be given an autonomous presentation.
- The standard presentation of Homotopy Type Theory given in the HoTT Book is not autonomous, since it depends upon *homotopy theory*.

An autonomous justification of path induction

Path induction follows from the uniqueness principle for identity types and Leibniz's law.

Interpreting HoTT

- The first thing we would need to do in giving an autonomous account of HoTT as a foundation is to give an interpretation – what exactly do we take types and tokens to be?
- The authors of the HoTT Book are not primarily interested in giving an account of the philosophical underpinnings of the theory, of course, so they don't go into detail on this.
- Two interpretations that they use:
  - types and tokens as *mathematical objects*
  - types and tokens as *spaces* and *points* (the homotopy interpretation)
- (Of course, we could also be Formalist and just explicitly deny that the formal language is to be interpreted at all – but where's the fun in that?)

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.
- It doesn't accord with the intensional treatment of types.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.
- It doesn't accord with the intensional treatment of types.
- Complex kinds aren't composed from simpler kinds.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.
- It doesn't accord with the intensional treatment of types.
- Complex kinds aren't composed from simpler kinds.
- Mathematical objects can belong to more than one kind.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.
- It doesn't accord with the intensional treatment of types.
- Complex kinds aren't composed from simpler kinds.
- Mathematical objects can belong to more than one kind.
- It reverses the relationship between types and tokens since objects are primary and kinds are secondary.

Tokens & Types as Mathematical Objects?

An obvious interpretation: tokens are mathematical objects, types are kinds of mathematical objects.

Problems with this interpretation:

- It commits us to Platonism.
- It doesn't accord with the intensional treatment of types.
- Complex kinds aren't composed from simpler kinds.
- Mathematical objects can belong to more than one kind.
- It reverses the relationship between types and tokens since objects are primary and kinds are secondary.
- It doesn't naturally account for the zero type.

Tokens & Types as Concepts

An alternative interpretation: types and tokens are concepts.

Specifically,

- types are **general concepts**;

Tokens & Types as Concepts

An alternative interpretation: types and tokens are concepts.

Specifically,

- types are **general concepts**;
- tokens of a type are **specific concepts** *qua* instance of that general concept.

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical
- It involves no extra ontological commitments.

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical
- It involves no extra ontological commitments.
- It accords with intensional type theory.

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical
- It involves no extra ontological commitments.
- It accords with intensional type theory.
- Complex concepts are composed from simpler concepts.

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical
- It involves no extra ontological commitments.
- It accords with intensional type theory.
- Complex concepts are composed from simpler concepts.
- It accords with each token belonging to exactly one type.

Advantages of the Types-as-Concepts Interpretation

- It is clearly pre-mathematical
- It involves no extra ontological commitments.
- It accords with intensional type theory.
- Complex concepts are composed from simpler concepts.
- It accords with each token belonging to exactly one type.
- It gets the order of dependence right: we must have the general concept before we can have a specific concept *qua* instance of the general concept

Ontology

- With a types-as-concepts interpretation we can give a minimal ontology for this foundation. We already have concepts as part of our everyday ontology, this isn't something we have to add for the purposes of doing mathematics.

Ontology

- With a types-as-concepts interpretation we can give a minimal ontology for this foundation. We already have concepts as part of our everyday ontology, this isn't something we have to add for the purposes of doing mathematics.
- This interpretation is compatible with the existence of mathematical objects but doesn't *require* them. We can have the concept of, say, the natural numbers without believing that there is an abstract object which the concept represents.

Ontology

- With a types-as-concepts interpretation we can give a minimal ontology for this foundation. We already have concepts as part of our everyday ontology, this isn't something we have to add for the purposes of doing mathematics.
- This interpretation is compatible with the existence of mathematical objects but doesn't *require* them. We can have the concept of, say, the natural numbers without believing that there is an abstract object which the concept represents.
- The foundation doesn't assert that any particular types exist (beyond the simple ones arising from the basic language). But if we posit the existence of, say, natural numbers, Hausdorff spaces, or non-principal ultrafilters, we can study them.

Epistemology

- We can know about concepts by thinking about them.

Epistemology

- We can know about concepts by thinking about them.
- The use of constructive logic guarantees that if we start with non-empty expressions for tokens and obey the rules of the theory we will only ever construct non-empty expressions for tokens.

Epistemology

- We can know about concepts by thinking about them.
- The use of constructive logic guarantees that if we start with non-empty expressions for tokens and obey the rules of the theory we will only ever construct non-empty expressions for tokens.
- Constructive logic does not require Brouwerian inutuitionism. We can work in a constructive logic without adopting or endorsing Brouwer's views on mathematical ontology.

Irrelevant distinctions

- Mathematicians abuse notation and collapse distinctions because in many cases these distinctions are irrelevant.

Irrelevant distinctions

- Mathematicians abuse notation and collapse distinctions because in many cases these distinctions are irrelevant.
- We are interested in *structural* properties – those that are preserved under structure-preserving maps – but we may be given a presentation of a structure that has other irrelevant properties.

Irrelevant distinctions

- Mathematicians abuse notation and collapse distinctions because in many cases these distinctions are irrelevant.
- We are interested in *structural* properties – those that are preserved under structure-preserving maps – but we may be given a presentation of a structure that has other irrelevant properties.
- As in category theory, different kinds of mathematical structures have corresponding kinds of structure-preserving maps.

Irrelevant distinctions

- Mathematicians abuse notation and collapse distinctions because in many cases these distinctions are irrelevant.
- We are interested in *structural* properties – those that are preserved under structure-preserving maps – but we may be given a presentation of a structure that has other irrelevant properties.
- As in category theory, different kinds of mathematical structures have corresponding kinds of structure-preserving maps.
- The Univalence axiom permits us to set aside the specific properties of any particular presentation in order to study the properties of the structure being presented because it allows us to identify types when there is the right kind of map between them.

Univalence

- The Univalence Axiom is an identity criterion for types. It says that *equivalent types are identical*. (For now, read 'equivalent' to mean 'isomorphic' – but we'll return to this later.)
- If two types are identical then trivially there is an equivalence between them. We have a function that maps any identification between types to an equivalence between those types.

$$\texttt{id-to-eq} : \texttt{Id}(\texttt{A}, \texttt{B}) \to \texttt{Equiv}(\texttt{A}, \texttt{B})$$

- Univalence posits a function in the opposite direction, mapping any equivalence between types to an identification between them.

$$\texttt{eq-to-id} : \texttt{Equiv}(\texttt{A}, \texttt{B}) \to \texttt{Id}(\texttt{A}, \texttt{B})$$

- So whenever we have an equivalence between types Univalence says we can replace it with an identity.

- More precisely, Univalence says for any types $\mathtt{A}$ and $\mathtt{B}$,

$$(\mathtt{A} \simeq \mathtt{B}) \quad \simeq \quad (\mathtt{A} = \mathtt{B})$$

The type of *equivalences between* A *and* B
is equivalent to
the type of *identifications of* A *and* B

- More precisely, Univalence says for any types $\mathtt{A}$ and $\mathtt{B}$,

$$(\mathtt{A} \simeq \mathtt{B}) \quad \simeq \quad (\mathtt{A} = \mathtt{B})$$

The type of *equivalences between* $\mathtt{A}$ *and* $\mathtt{B}$
is equivalent to
the type of *identifications of* $\mathtt{A}$ *and* $\mathtt{B}$

- Note that this doesn't follow from the basic setup of HoTT – if we want Univalence we must add it as an axiom. Most people currently working with HoTT adopt Univalence without question.

Universes and Univalence

In fact we cannot quantify over all types.

In fact we cannot quantify over all types.

Types live in *universes* and we can only quantify over all the types in a universe.

In fact we cannot quantify over all types.

Types live in *universes* and we can only quantify over all the types in a universe.

'Typical Ambiguity'

In fact we cannot quantify over all types.

Types live in *universes* and we can only quantify over all the types in a universe.

'Typical Ambiguity'

The meaning of univalence

Universes and Univalence

In fact we cannot quantify over all types.

Types live in *universes* and we can only quantify over all the types in a universe.

'Typical Ambiguity'

The meaning of univalence

Extending the concepts interpretation of types to universes.

- Rather than abusing notation to surreptitiously ignore the distinction between equivalent objects, Univalence permits us to explicitly identify them in a rigorous way within the system.

Using Univalence

- Rather than abusing notation to surreptitiously ignore the distinction between equivalent objects, Univalence permits us to explicitly identify them in a rigorous way within the system.
- We are then free to move between different presentations of the same structure without having the details of these presentations obscure the underlying structure.

Using Univalence

- Rather than abusing notation to surreptitiously ignore the distinction between equivalent objects, Univalence permits us to explicitly identify them in a rigorous way within the system.
- We are then free to move between different presentations of the same structure without having the details of these presentations obscure the underlying structure.
- What would normally be achieved by ‘abuses of notation’ can be officially sanctioned and carried out formally in a way that a proof checker can understand.

Using Univalence

- Rather than abusing notation to surreptitiously ignore the distinction between equivalent objects, Univalence permits us to explicitly identify them in a rigorous way within the system.
- We are then free to move between different presentations of the same structure without having the details of these presentations obscure the underlying structure.
- What would normally be achieved by 'abuses of notation' can be officially sanctioned and carried out formally in a way that a proof checker can understand.
- This is one way in which Homotopy Type Theory closes the gap between foundations and practice.

- Steve Awodey claims that a structuralist view of mathematics motivates Univalence.

- Steve Awodey claims that a structuralist view of mathematics motivates Univalence.
- An equivalence between types means that they're really just different presentations of some common structure.

Univalence and Structuralism

- Steve Awodey claims that a structuralist view of mathematics motivates Univalence.
- An equivalence between types means that they're really just different presentations of some common structure.
- The Invariance Principle (IP): reasoning in the system should be invariant under (the appropriate notion of) equivalence, so that we are reasoning about the underlying structures themselves, not about some particular presentation.

Univalence and Structuralism

- Steve Awodey claims that a structuralist view of mathematics motivates Univalence.
- An equivalence between types means that they're really just different presentations of some common structure.
- The Invariance Principle (IP): reasoning in the system should be invariant under (the appropriate notion of) equivalence, so that we are reasoning about the underlying structures themselves, not about some particular presentation.
- Thus any property satisfied by some type `A` should be satisfied by any type `B` that is equivalent to `A`. But if we consider the property of *being identical to* `A`, Univalence follows. So IP is equivalent to Univalence.

What is 'Equivalence'?

- Recall the statement of Univalence:

$$(\mathtt{A} \simeq \mathtt{B}) \quad \simeq \quad (\mathtt{A} = \mathtt{B})$$

where $\simeq$ denotes 'equivalence' between types. What exactly is this relation?

What is 'Equivalence'?

- Recall the statement of Univalence:

$$(\mathtt{A} \simeq \mathtt{B}) \quad \simeq \quad (\mathtt{A} = \mathtt{B})$$

  where $\simeq$ denotes 'equivalence' between types. What exactly is this relation?

- If Univalence is motivated by structuralism as Awodey suggests, then it would be reasonable to assume that $\simeq$ is isomorphism.

What is 'Equivalence'?

- Recall the statement of Univalence:

$$(\mathtt{A} \simeq \mathtt{B}) \simeq (\mathtt{A} = \mathtt{B})$$

  where $\simeq$ denotes 'equivalence' between types. What exactly is this relation?

- If Univalence is motivated by structuralism as Awodey suggests, then it would be reasonable to assume that $\simeq$ is isomorphism.
- However, this is not correct! Specifically, this version of Univalence with $\simeq$ understood as isomorphism (which we call 'Isovalence') can be proved to be *inconsistent*.

What is 'Equivalence'?

- Recall the statement of Univalence:

$$(\mathtt{A} \simeq \mathtt{B}) \quad \simeq \quad (\mathtt{A} = \mathtt{B})$$

  where $\simeq$ denotes 'equivalence' between types. What exactly is this relation?

- If Univalence is motivated by structuralism as Awodey suggests, then it would be reasonable to assume that $\simeq$ is isomorphism.
- However, this is not correct! Specifically, this version of Univalence with $\simeq$ understood as isomorphism (which we call 'Isovalence') can be proved to be *inconsistent*.
- Thus if Univalence is to be consistent, $\simeq$ must be some other relation.

Bi-invertibility

- A function $\mathtt{f} : \mathtt{A} \to \mathtt{B}$ is an **isomorphism** iff there is a function $\mathtt{g} : \mathtt{B} \to \mathtt{A}$ such that

$$\mathtt{g}(\mathtt{f}(\mathtt{a})) = \mathtt{a} \quad \text{for every } \mathtt{a} : \mathtt{A}$$
$$\mathtt{f}(\mathtt{g}(\mathtt{b})) = \mathtt{b} \quad \text{for every } \mathtt{b} : \mathtt{B}$$

i.e. there is a single function that is both a left- and right-inverse.

Bi-invertibility

- A function $\mathtt{f} : \mathtt{A} \to \mathtt{B}$ is an **isomorphism** iff there is a function $\mathtt{g} : \mathtt{B} \to \mathtt{A}$ such that

$$\mathtt{g}(\mathtt{f}(\mathtt{a})) = \mathtt{a} \quad \text{for every } \mathtt{a} : \mathtt{A}$$
$$\mathtt{f}(\mathtt{g}(\mathtt{b})) = \mathtt{b} \quad \text{for every } \mathtt{b} : \mathtt{B}$$

i.e. there is a single function that is both a left- and right-inverse.

- A function $\mathtt{f} : \mathtt{A} \to \mathtt{B}$ is **bi-invertible** iff there are functions $\mathtt{g}, \mathtt{h} : \mathtt{B} \to \mathtt{A}$ such that

$$\mathtt{g}(\mathtt{f}(\mathtt{a})) = \mathtt{a} \quad \text{for every } \mathtt{a} : \mathtt{A}$$
$$\mathtt{f}(\mathtt{h}(\mathtt{b})) = \mathtt{b} \quad \text{for every } \mathtt{b} : \mathtt{B}$$

i.e. there is a left-inverse and right-inverse, but they're not required to be the same function.

The definition of equivalence

- If we define equivalence to mean the existence of a bi-invertible function then Univalence is consistent.

The definition of equivalence

- If we define equivalence to mean the existence of a bi-invertible function then Univalence is consistent.
- There are other definitions of equivalence that also make Univalence consistent.

The definition of equivalence

- If we define equivalence to mean the existence of a bi-invertible function then Univalence is consistent.
- There are other definitions of equivalence that also make Univalence consistent.
- Any of these are equally good choices for the definition of equivalence: specifically, on the assumption of any consistent ‘flavour’ of Univalence, we can prove that these notions of equivalence are all equivalent to one another.

The definition of equivalence

- If we define equivalence to mean the existence of a bi-invertible function then Univalence is consistent.
- There are other definitions of equivalence that also make Univalence consistent.
- Any of these are equally good choices for the definition of equivalence: specifically, on the assumption of any consistent 'flavour' of Univalence, we can prove that these notions of equivalence are all equivalent to one another.
- But isomorphism doesn't fit into this pattern: it's not equivalent to bi-invertibility (or the other definitions).

- If we define equivalence to mean the existence of a bi-invertible function then Univalence is consistent.
- There are other definitions of equivalence that also make Univalence consistent.
- Any of these are equally good choices for the definition of equivalence: specifically, on the assumption of any consistent ‘flavour’ of Univalence, we can prove that these notions of equivalence are all equivalent to one another.
- But isomorphism doesn’t fit into this pattern: it’s not equivalent to bi-invertibility (or the other definitions).
- What’s the difference?

Isomorphism vs Equivalence

- The difference between Isomorphism and Equivalence is subtle!

Isomorphism vs Equivalence

- The difference between Isomorphism and Equivalence is subtle!
- Two types that are isomorphic are also equivalent, and vice versa

$$\texttt{Iso}(\texttt{A}, \texttt{B}) \leftrightarrow \texttt{Equiv}(\texttt{A}, \texttt{B})$$

Isomorphism vs Equivalence

- The difference between Isomorphism and Equivalence is subtle!
- Two types that are isomorphic are also equivalent, and vice versa

$$\texttt{Iso}(\texttt{A}, \texttt{B}) \leftrightarrow \texttt{Equiv}(\texttt{A}, \texttt{B})$$

- If a function `f` has both a left-inverse `g` and a right-inverse `h` (i.e. it's bi-invertible) then we can prove that both `g` and `h` are full inverses of `f` – i.e. either of them is sufficient to prove that `f` is an isomorphism.

The difference between Iso and Equiv

- So the difference between $\texttt{Iso}(\texttt{A},\texttt{B})$ and $\texttt{Equiv}(\texttt{A},\texttt{B})$ is a proof-relevant distinction: it's about the specific evidence that's required in order to demonstrate that two types stand in that particular relation.

The difference between Iso and Equiv

- So the difference between $\mathtt{Iso}(\mathtt{A}, \mathtt{B})$ and $\mathtt{Equiv}(\mathtt{A}, \mathtt{B})$ is a proof-relevant distinction: it's about the specific evidence that's required in order to demonstrate that two types stand in that particular relation.
- $\mathtt{Iso}(\mathtt{A}, \mathtt{B})$ requires a function f, an inverse g, and a certificate that g is inverse to f.

The difference between Iso and Equiv

- So the difference between `Iso(A, B)` and `Equiv(A, B)` is a proof-relevant distinction: it's about the specific evidence that's required in order to demonstrate that two types stand in that particular relation.
- `Iso(A, B)` requires a function `f`, an inverse `g`, and a certificate that `g` is inverse to `f`.
- `Equiv(A, B)` requires a function `f`, and functions `g` and `h`, and certificates that they are respectively left- and right-inverse to `f`.

The difference between Iso and Equiv

- So the difference between $\texttt{Iso}(\texttt{A}, \texttt{B})$ and $\texttt{Equiv}(\texttt{A}, \texttt{B})$ is a proof-relevant distinction: it's about the specific evidence that's required in order to demonstrate that two types stand in that particular relation.
- $\texttt{Iso}(\texttt{A}, \texttt{B})$ requires a function `f`, an inverse `g`, and a certificate that `g` is inverse to `f`.
- $\texttt{Equiv}(\texttt{A}, \texttt{B})$ requires a function `f`, and functions `g` and `h`, and certificates that they are respectively left- and right-inverse to `f`.
- But why should this difference be so important – important enough that Univalence is consistent while Isovalence is inconsistent?

THE DIFFERENCE BETWEEN Iso AND Equiv

- So the difference between Iso(A, B) and Equiv(A, B) is a proof-relevant distinction: it's about the specific evidence that's required in order to demonstrate that two types stand in that particular relation.
- Iso(A, B) requires a function f, an inverse g, and a certificate that g is inverse to f.
- Equiv(A, B) requires a function f, and functions g and h, and certificates that they are respectively left- and right-inverse to f.
- But why should this difference be so important – important enough that Univalence is consistent while Isovalence is inconsistent?
- In one sense equivalence is weaker than isomorphism, but in another sense it is stronger: equivalence of types is a *mere proposition*.

Justifying Univalence

- Most of the use that is made of HoTT in mathematical practice is of Univalent HoTT. This is exemplified by the fact that it is introduced early in the HoTT-Book and is then used throughout the remainder.

- Most of the use that is made of HoTT in mathematical practice is of Univalent HoTT. This is exemplified by the fact that it is introduced early in the HoTT-Book and is then used throughout the remainder.
- UA entails function extensionality

Justifying Univalence

- Most of the use that is made of HoTT in mathematical practice is of Univalent HoTT. This is exemplified by the fact that it is introduced early in the HoTT-Book and is then used throughout the remainder.
- UA entails function extensionality
- Arguably the reason Univalence is useful is that it represents how mathematicians think by identifying types that would otherwise be kept distinct. For example, in HoTT as in set theory there are different ways to define ordered pairs, but in Univalent HoTT (unlike in set theory) alternative equivalent types of ordered pairs are identified.

Understanding Univalence

Univalence is best understood as the commitment to work in univalent universes because there is no reason to believe that there are not non-univalent universes.

Understanding Univalence

Univalence is best understood as the commitment to work in univalent universes because there is no reason to believe that there are not non-univalent universes.

Univalence is consistent because we can always infer `P(B)` from `P(A)` if `A` and `B` are equivalent.

In HoTT without UA there is nothing we can say in the language about one of two equivalent types that we cannot say about the other but we cannot say they are identical.

Understanding Univalence

Univalence is best understood as the commitment to work in univalent universes because there is no reason to believe that there are not non-univalent universes.

Univalence is consistent because we can always infer `P(B)` from `P(A)` if `A` and `B` are equivalent.

In HoTT without UA there is nothing we can say in the language about one of two equivalent types that we cannot say about the other but we cannot say they are identical.

Examples of the meaning of UA

To see what it means to say that equivalent types are identical consider some examples:

(i) $\mathtt{A} \times \mathtt{B}$ and $\mathtt{B} \times \mathtt{A}$ are (externally) distinct types, but in almost any context in which we are interested in them there is no effective difference between them, so it makes sense to equate them. We can think of this as creating the type 'unordered pair'.

Examples of the meaning of UA

To see what it means to say that equivalent types are identical consider some examples:

(i) $\texttt{A} \times \texttt{B}$ and $\texttt{B} \times \texttt{A}$ are (externally) distinct types, but in almost any context in which we are interested in them there is no effective difference between them, so it makes sense to equate them. We can think of this as creating the type 'unordered pair'.

(ii) The list-sorting algorithms MergeSort and InsertionSort. Clearly these are distinct algorithms, and in some contexts the differences between them (e.g. their running times on a given list) are important. But in another sense, regarded just as relations between inputs and outputs, they are identical since they produce the same output when given the same input.

Examples of the meaning of UA

To see what it means to say that equivalent types are identical consider some examples:

(I) $\mathtt{A} \times \mathtt{B}$ and $\mathtt{B} \times \mathtt{A}$ are (externally) distinct types, but in almost any context in which we are interested in them there is no effective difference between them, so it makes sense to equate them. We can think of this as creating the type 'unordered pair'.

(II) The list-sorting algorithms MergeSort and InsertionSort. Clearly these are distinct algorithms, and in some contexts the differences between them (e.g. their running times on a given list) are important. But in another sense, regarded just as relations between inputs and outputs, they are identical since they produce the same output when given the same input.

(III) All empty types, for example, *even divisors of 9* and *largest prime*, are equivalent to $\mathtt{0}$ and thus (under univalence) identical to it.

- Univalence allows us to exploit the benefits of the intensional nature of HoTT – viz. non-trivial identities and higher identity structure – whilst recovering the extensional character of much mathematical thought.

- Univalence allows us to exploit the benefits of the intensional nature of HoTT – viz. non-trivial identities and higher identity structure – whilst recovering the extensional character of much mathematical thought.
- The language of HoTT without Univalence is so fine-grained as to distinguish types that differ only in how they are described. Univalence allows us to identify such types.

- Univalence allows us to exploit the benefits of the intensional nature of HoTT – viz. non-trivial identities and higher identity structure – whilst recovering the extensional character of much mathematical thought.
- The language of HoTT without Univalence is so fine-grained as to distinguish types that differ only in how they are described. Univalence allows us to identify such types.
- Purely intensional theories such as basic HoTT make distinctions that are too fine-grained for ordinary mathematical practice.

- Univalence allows us to exploit the benefits of the intensional nature of HoTT – viz. non-trivial identities and higher identity structure – whilst recovering the extensional character of much mathematical thought.
- The language of HoTT without Univalence is so fine-grained as to distinguish types that differ only in how they are described. Univalence allows us to identify such types.
- Purely intensional theories such as basic HoTT make distinctions that are too fine-grained for ordinary mathematical practice.
- Purely extensional theories such as set theory collapses some of these distinctions, but introduces unwanted distinctions between different ways of representing mathematical structures (such as ordered pairs).

- Univalence allows us to exploit the benefits of the intensional nature of HoTT – viz. non-trivial identities and higher identity structure – whilst recovering the extensional character of much mathematical thought.
- The language of HoTT without Univalence is so fine-grained as to distinguish types that differ only in how they are described. Univalence allows us to identify such types.
- Purely intensional theories such as basic HoTT make distinctions that are too fine-grained for ordinary mathematical practice.
- Purely extensional theories such as set theory collapses some of these distinctions, but introduces unwanted distinctions between different ways of representing mathematical structures (such as ordered pairs).
- Univalence strikes a balance between the two, introducing an element of extensionality into the intensional theory of HoTT.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.
- We can consider identifications (and relations between them) as objects of study in their own right.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.
- We can consider identifications (and relations between them) as objects of study in their own right.
- HoTT makes a clear distinction between *definition* and *existence*: we say what it is to be an X by constructing a corresponding type;
  we say that an X exists by producing a token of that type.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.
- We can consider identifications (and relations between them) as objects of study in their own right.
- HoTT makes a clear distinction between *definition* and *existence*: we say what it is to be an X by constructing a corresponding type;
  we say that an X exists by producing a token of that type.
- Even in HoTT with UA identity is not replaced by equivalence indeed the latter is defined in terms of the former. Identity is primitive in HoTT.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.
- We can consider identifications (and relations between them) as objects of study in their own right.
- HoTT makes a clear distinction between *definition* and *existence*: we say what it is to be an X by constructing a corresponding type;
  we say that an X exists by producing a token of that type.
- Even in HoTT with UA identity is not replaced by equivalence indeed the latter is defined in terms of the former. Identity is primitive in HoTT.
- Equivalence is not defined for tokens that are not themselves types but only for types.

Conclusions

- Equipped with the Types-as-Concepts interpretation, HoTT is a candidate foundation for mathematics.
- UA expresses a key insight of mathematical structuralism.
- We can consider identifications (and relations between them) as objects of study in their own right.
- HoTT makes a clear distinction between *definition* and *existence*: we say what it is to be an X by constructing a corresponding type;
  we say that an X exists by producing a token of that type.
- Even in HoTT with UA identity is not replaced by equivalence indeed the latter is defined in terms of the former. Identity is primitive in HoTT.
- Equivalence is not defined for tokens that are not themselves types but only for types.

THANK YOU

- http://bristol.ac.uk/homotopy-type-theory
- A Primer on Homotopy Type Theory. www.bristol.ac.uk/homotopy-type-theory/, 2014.
- Does Homotopy Type Theory Provide a Foundation for Mathematics? *BJPS*, forthcoming.
- Identity in Homotopy Type Theory, Part I: The Justification of Path Induction. *Phil Math*, 23:386–406, 2015.
- Identity in Homotopy Type Theory: Part II, The Conceptual and Philosophical Status of Identity in HoTT. *Phil Math*.
- Universes and Univalence in Homotopy Type Theory. forthcoming, *The Review of Symbolic Logic*.
- Representation and Symmetry in Physics. Under review, *Phil Sci*.