An introduction to Homotopy Type Theory

Dan Saattrup Nielsen & Martin Speirs

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What's going to happen?

- What is homotopy type theory?
- Basic type theory
- A proof of the axiom of choice
- Isomorphic objects are equal
- Higher inductive types
- Fundamental group of the circle is the integers

What is homotopy type theory?

- It's a *synthetic* approach to homotopy theory
- It's a *foundational* theory
- Connections with category theory, topology and logic
- Propositions as types
- HoTT = ITT + UA + HITs

What is homotopy type theory?

- It's a *synthetic* approach to homotopy theory
- It's a *foundational* theory
- Connections with category theory, topology and logic
- Propositions as types
- HoTT = ITT + UA + HITs

What is homotopy type theory?

- It's a *synthetic* approach to homotopy theory
- It's a *foundational* theory
- Connections with category theory, topology and logic
- Propositions as types
- HoTT = ITT + UA + HITs

What is homotopy type theory?

- It's a *synthetic* approach to homotopy theory
- It's a *foundational* theory
- Connections with category theory, topology and logic
- Propositions as types
- HoTT = ITT + UA + HITs

What is homotopy type theory?

- It's a *synthetic* approach to homotopy theory
- It's a *foundational* theory
- Connections with category theory, topology and logic
- Propositions as types
- HoTT $=$ ITT $+$ UA $+$ HITs

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, $\ldots$
- Type theory builds on *rules* rather than *axioms*

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
    - Types A
    - Terms $a : A$
    - Dependent types $A(x)$
    - Dependent terms $a(x) : A(x)$
    - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, $\ldots$
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, $\ldots$
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
    - Types A
    - Terms $a : A$
    - Dependent types $A(x)$
    - Dependent terms $a(x) : A(x)$
    - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, ...
- Type theory builds on *rules* rather than *axioms*

Type theory

- Type theory is *syntactical*
- Type theory is an *independent* foundation - no logical foundation is needed
- Type theory structure:
  - Types A
  - Terms $a : A$
  - Dependent types $A(x)$
  - Dependent terms $a(x) : A(x)$
  - $A \times B$, $A \to B$, $\Sigma_{(a:A)} B(a)$, $\Pi_{(a:A)} B(a)$, $\ldots$
- Type theory builds on *rules* rather than *axioms*

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{"}g \text{ function"} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$. □

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{“}g \text{ function”} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$. $\square$

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{“}g \text{ function”} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x,X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X),X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathrm{fst} f(X), \lambda X.\mathrm{snd} f(X)\rangle$. $\square$

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{"}g \text{ function"} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$. □

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{“}g \text{ function”} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x,X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X),X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$. $\square$

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{"}g \text{ function"} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathrm{fst} f(X), \lambda X.\mathrm{snd} f(X)\rangle$.

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{“}g \text{ function”} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathrm{fst} f(X), \lambda X.\mathrm{snd} f(X)\rangle$.

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{"}g \text{ function"} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$.

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{"}g \text{ function"} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x,X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X),X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathrm{fst} f(X), \lambda X.\mathrm{snd} f(X)\rangle$.

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)($"g function" $\wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x,X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X),X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathrm{fst} f(X), \lambda X.\mathrm{snd} f(X)\rangle$.

Axiom of choice

- Regular AC: "Given a family of non-empty sets F, there exists a choice function g with domain F, satisfying $g(X) \in X$ for all $X \in F$."
- In regular logic:
  $(\forall X \in F)(\exists x \in X) \Rightarrow$
  $(\exists g)(\text{“}g \text{ function”} \wedge \operatorname{dom} g = F \wedge (\forall X \in F)(g(X) \in X))$

Theorem of Choice

$$\left(\prod_{(X:F)} \sum_{(x:X)} P(x, X)\right) \to \left(\sum_{(g:\prod_{(X:F)} X)} \prod_{(X:F)} P(g(X), X)\right)$$

Proof

Take $\lambda f.\langle \lambda X.\mathsf{fst} f(X), \lambda X.\mathsf{snd} f(X)\rangle$. □

The Identity type

- The identity type $Id_A(x, y)$ (also written $x =_A y$) for $x, y : A$
- $p : Id_A(x, y)$ is a *proof* of $x = y$, or a *path* from x to y
- Special element, $refl_x : Id_A(x, x)$ ("constant path at x")
- What is $Id_{Id_A}(p, q)$?

- The identity type $Id_A(x, y)$ (also written $x =_A y$) for $x, y : A$
- $p : Id_A(x, y)$ is a *proof* of $x = y$, or a *path* from x to y
- Special element, $refl_x : Id_A(x, x)$ ("constant path at x")
- What is $Id_{Id_A}(p, q)$?

- The identity type $Id_A(x, y)$ (also written $x =_A y$) for $x, y : A$
- $p : Id_A(x, y)$ is a *proof* of $x = y$, or a *path* from x to y
- Special element, $refl_x : Id_A(x, x)$ ("constant path at x")
- What is $Id_{Id_A}(p, q)$?

The Identity type

- The identity type $Id_A(x, y)$ (also written $x =_A y$) for $x, y : A$
- $p : Id_A(x, y)$ is a *proof* of $x = y$, or a *path* from x to y
- Special element, $refl_x : Id_A(x, x)$ ("constant path at x")
- What is $Id_{Id_A}(p, q)$?

The Identity type

- The identity type $Id_A(x, y)$ (also written $x =_A y$) for $x, y : A$
- $p : Id_A(x, y)$ is a *proof* of $x = y$, or a *path* from x to y
- Special element, $refl_x : Id_A(x, x)$ ("constant path at x")
- What is $Id_{Id_A}(p, q)$?

Univalence

- Isomorphism type $X \cong Y$ for X, Y types:
$$\sum_{(f:X\to Y)} \sum_{(g:Y\to X)} \left(\prod_{(x:X)} g(f(x)) = x\right) \times \left(\prod_{(y:Y)} f(g(y)) = y\right)$$
- $p : X \cong Y$ is a *proof* of $X \cong Y$

Univalence axiom

For all types X, Y it holds that $(X = Y) \cong (X \cong Y)$; i.e. "isomorphic objects are equal".

- Isomorphism type $X \cong Y$ for X, Y types:
$$\sum_{(f:X\to Y)} \sum_{(g:Y\to X)} \left(\prod_{(x:X)} g(f(x)) = x\right) \times \left(\prod_{(y:Y)} f(g(y)) = y\right)$$
- $p : X \cong Y$ is a *proof* of $X \cong Y$

Univalence axiom

For all types X, Y it holds that $(X = Y) \cong (X \cong Y)$; i.e. "isomorphic objects are equal".

Univalence

- Isomorphism type $X \cong Y$ for X, Y types:
  $\sum_{(f:X\to Y)} \sum_{(g:Y\to X)}$
  $\left(\prod_{(x:X)} g(f(x)) = x\right) \times \left(\prod_{(y:Y)} f(g(y)) = y\right)$
- $p : X \cong Y$ is a *proof* of $X \cong Y$

Univalence axiom

For all types X, Y it holds that $(X = Y) \cong (X \cong Y)$; i.e. "isomorphic objects are equal".

Univalence

- Isomorphism type $X \cong Y$ for X, Y types: $\sum_{(f:X\to Y)} \sum_{(g:Y\to X)} \left(\prod_{(x:X)} g(f(x)) = x\right) \times \left(\prod_{(y:Y)} f(g(y)) = y\right)$
- $p : X \cong Y$ is a *proof* of $X \cong Y$

Univalence axiom

For all types X, Y it holds that $(X = Y) \cong (X \cong Y)$; i.e. "isomorphic objects are equal".