Martin Löf Type Theory: Univalence implies function extensionality

Thomas Pipilikas

INTER-INSTITUTIONAL GRADUATE PROGRAM "ALGORITHMS, LOGIC AND DISCRETE MATHEMATICS"

Disclaimer

We will not use the notion of μετασχηματισμού.

Disclaimer

We will not use the notion of μετασχηματισμού.
Here we include the uniqueness principle

$$f \equiv \lambda x.f\,(x)$$

as a principle of judgmental equality.

Disclaimer

We will not use the notion of μετασχηματισμού.
Here we include the uniqueness principle

$$f \equiv \lambda x.f\left(x\right)$$

as a principle of judgmental equality.
vs η-conversion (Λήμμα 8.) Για οποιαδήποτε $f : A \to B$ υπάρχει ένα

$$\eta_{A\to B}\left(f\right) : \lambda\left(\mathsf{apply}_f\right) = f$$

το οποίο ικανοποιεί τη σχέση

$$\eta_{A\to B}\left(\lambda\left(b\right)\right) \equiv \mathsf{refl}_{\lambda(b)},$$

όπου $\left(x : A\right) b\left(x\right) : B$

Disclaimer

I know nothing on Homotopy Theory...

I'll just be a coq!

Disclaimer

I know nothing on Homotopy Theory...

I'll just be a coq!

"there they laugh: they do not understand me; I am not the mouth for these ears."

Also sprach Zarathustra: Ein Buch für Alle und Keinen, Friedrich Nietzsche

Why HoTT is hot?!

- 1986 *"Algebraic Cycles and Higher K-theory"* , by Spencer Bloch contained a mistake (Lemma 1.1).

Why HoTT is hot?!

- 1986 *"Algebraic Cycles and Higher K-theory"* , by Spencer Bloch contained a mistake (Lemma 1.1).
- 1989 *"∞-groupoids as a model for a homotopy category"* , by Michael Kapranov and Vladimir Voevodsky.

Why HoTT is hot?!

- 1986 *"Algebraic Cycles and Higher K-theory"* , by Spencer Bloch contained a mistake (Lemma 1.1).
- 1989 *"∞-groupoids as a model for a homotopy category"* , by Michael Kapranov and Vladimir Voevodsky.
- 1993 A new proof of Lemma 1.1! It took many more years for it to be accepted as correct.

Why HoTT is hot?!

- 1986 *"Algebraic Cycles and Higher K-theory"* , by Spencer Bloch contained a mistake (Lemma 1.1).
- 1989 *"∞-groupoids as a model for a homotopy category"* , by Michael Kapranov and Vladimir Voevodsky.
- 1993 A new proof of Lemma 1.1! It took many more years for it to be accepted as correct.
- 1992/1993 *"Cohomological Theory of Presheaves with Transfers."* , by Vladimir Voevodsky.

Why HoTT is hot?!

- 1986 *"Algebraic Cycles and Higher K-theory"* , by Spencer Bloch contained a mistake (Lemma 1.1).
- 1989 *"∞-groupoids as a model for a homotopy category"* , by Michael Kapranov and Vladimir Voevodsky.
- 1993 A new proof of Lemma 1.1! It took many more years for it to be accepted as correct.
- 1992/1993 *"Cohomological Theory of Presheaves with Transfers."* , by Vladimir Voevodsky.

  The approach to Motivic Cohomology circumvented Bloch's lemma by relying on this paper.

Why HoTT is hot?!

- 1999/2000 *"Cohomological Theory of Presheaves with Transfers."* contained a mistake!!!

Why HoTT is hot?!

- 1999/2000 *"Cohomological Theory of Presheaves with Transfers."* contained a mistake!!!

  This story got me scared. Starting from 1993 multiple groups of mathematicians studied the "Cohomological Theory" paper at seminars and used it in their work and none of them noticed the mistake.

  Vladimir Voevodsky

Why HoTT is hot?!

- 1999/2000 *"Cohomological Theory of Presheaves with Transfers."* contained a mistake!!!

  This story got me scared. Starting from 1993 multiple groups of mathematicians studied the "Cohomological Theory" paper at seminars and used it in their work and none of them noticed the mistake.

  Vladimir Voevodsky

- 1998 *"Homotopy types of strict 3-groupoids"* , by Carlos Simpson contained a counter example on "∞-groupoids as a model for a homotopy category" paper.

Why HoTT is hot?!

- Mathematical research currently relies on a complex system of mutual trust based on reputations.

Why HoTT is hot?!

- Mathematical research currently relies on a complex system of mutual trust based on reputations.
- We are off to uncharted waters!

Why HoTT is hot?!

- Mathematical research currently relies on a complex system of mutual trust based on reputations.
- We are off to uncharted waters!

> *the only real long-term solution to the problems that I encountered is to start using computers in the verification of mathematical reasoning.*
>
> *Vladimir Voevodsky[3]*

Why HoTT is hot?!

- We need new proof verifiers!

Why HoTT is hot?!

- We need new proof verifiers!

The roadblock that prevented generations of interested mathematicians and computer scientists from solving the problem of computer verification of mathematical reasoning was the unpreparedness of foundations of mathematics for the requirements of this task.

Vladimir Voevodsky[3]

Why HoTT is hot?!

- Why Type Theory?!

Why HoTT is hot?!

- Why Type Theory?!

…Ένα πλεονέκτημα της χρήσης τής θεωρίας τύπων για την κατασκευή προγραμμάτων είναι ότι είναι δυνατόν να εκφράσουμε τόσο τις προδιαγραφές όσο και τα προγράμματα μέσα στον ίδιο φορμαλισμό….

Νικόλας Ρήγας

Why HoTT is hot?!

- Why Type Theory?!

  …Ένα πλεονέκτημα της χρήσης τής θεωρίας τύπων για την κατασκευή προγραμμάτων είναι ότι είναι δυνατόν να εκφράσουμε τόσο τις προδιαγραφές όσο και τα προγράμματα μέσα στον ίδιο φορμαλισμό….

  Νικόλας Ρήγας

- The idea of Homotopy Type Theory arose around 2006 in independent work by Awodey and Warren and Voevodsky, but it was inspired by Hofmann and Streicher's earlier groupoid interpretation [2].

Why HoTT is hot?!

- Why Type Theory?!

  ...Ένα πλεονέκτημα της χρήσης τής θεωρίας τύπων για την κατασκευή προγραμμάτων είναι ότι είναι δυνατόν να εκφράσουμε τόσο τις προδιαγραφές όσο και τα προγράμματα μέσα στον ίδιο φορμαλισμό....

  Νικόλας Ρήγας

- The idea of Homotopy Type Theory arose around 2006 in independent work by Awodey and Warren and Voevodsky, but it was inspired by Hofmann and Streicher's earlier groupoid interpretation [2].
- In particular, Voevodsky constructed an interpretation of type theory in Kan simplicial sets, and recognized that this interpretation satisfied a further crucial property which he dubbed **univalence**.

Why HoTT is hot?!

- Coq, Agda...

Why HoTT is hot?!

- Coq, Agda...

    - The first such library called *"Foundations"* was created by Vladimir Voevodsky in 2010.

Why HoTT is hot?!

- Coq, Agda...
    - The first such library called *"Foundations"* was created by Vladimir Voevodsky in 2010.
    - HoTT Coq library and HoTT Agda library.

Why HoTT is hot?!

- Coq, Agda...
    - The first such library called *"Foundations"* was created by Vladimir Voevodsky in 2010.
    - HoTT Coq library and HoTT Agda library.

...many of the proofs described in this book (HoTT) were actually first done in a fully formalized form in a proof assistant, and are only now being "unformalized" for the first time — a reversal of the usual relation between formal and informal mathematics. [1]

Homotopy

Definition

Let $f, g : \prod_{(x:A)} P(x)$ be two sections of a type family $P : A \to \mathcal{U}$. A **homotopy** from f to g is a dependent function of type

$$(f \sim g) :\equiv \prod_{x:A} (f(x) = g(x)).$$

Homotopy

Definition

Let $f, g : \prod_{(x:A)} P(x)$ be two sections of a type family $P : A \to \mathcal{U}$. A **homotopy** from f to g is a dependent function of type

$$(f \sim g) :\equiv \prod_{x:A} (f(x) = g(x)).$$

Lemma (Lemma 2.4.3.)

Suppose $H : f \sim g$ *is a homotopy between functions* $f, g : A \to B$ *and let* $p : x =_A y$*. Then we have*

$$H(x) \cdot g(p) = f(p) \cdot H(y).$$

We may also draw this as a commutative diagram:

$$\begin{tikzcd}
f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\
g(x) \arrow[r, equal, "g(p)"'] & g(y)
\end{tikzcd}$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.

We may also draw this as a commutative diagram:

$$\begin{tikzcd} f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\ g(x) \arrow[r, equal, "g(p)"'] & g(y) \end{tikzcd}$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.Then trivially we observe that

$$H(x) \cdot g(\mathsf{refl}_x) = f(\mathsf{refl}_x) \cdot H(x)$$

We may also draw this as a commutative diagram:

$$
\begin{tikzcd}
f(x) \arrow[r, equals, "f(p)"] \arrow[d, equals, "H(x)"'] & f(y) \arrow[d, equals, "H(y)"] \\
g(x) \arrow[r, equals, "g(p)"'] & g(y)
\end{tikzcd}
$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.Then trivially we observe that

$$H(x) \cdot g(\mathsf{refl}_x) = f(\mathsf{refl}_x) \cdot H(x) \;:\equiv H(x) \cdot \mathsf{ap}_g(\mathsf{refl}_x) = \mathsf{ap}_f(\mathsf{refl}_x) \cdot H(x)$$

We may also draw this as a commutative diagram:

$$\begin{tikzcd}
f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\
g(x) \arrow[r, equal, "g(p)"'] & g(y)
\end{tikzcd}$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.Then trivially we observe that

$$\begin{aligned} H(x) \cdot g(\mathsf{refl}_x) = f(\mathsf{refl}_x) \cdot H(x) \; &:\equiv H(x) \cdot \mathsf{ap}_g\ (\mathsf{refl}_x) = \mathsf{ap}_f\ (\mathsf{refl}_x) \cdot H(x) \\ &:\equiv H(x) \cdot \mathsf{refl}_x = \mathsf{refl}_x \cdot H(x) \end{aligned}$$

We may also draw this as a commutative diagram:

$$\begin{tikzcd} f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\ g(x) \arrow[r, equal, "g(p)"'] & g(y) \end{tikzcd}$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.Then trivially we observe that

$$\begin{aligned} H(x) \cdot g(\mathsf{refl}_x) = f(\mathsf{refl}_x) \cdot H(x) \; &:\equiv H(x) \cdot \mathsf{ap}_g(\mathsf{refl}_x) = \mathsf{ap}_f(\mathsf{refl}_x) \cdot H(x) \\ &:\equiv H(x) \cdot \mathsf{refl}_x = \mathsf{refl}_x \cdot H(x) \\ &:\equiv H(x) = H(x) \end{aligned}$$

We may also draw this as a commutative diagram:

$$\begin{tikzcd}
f(x) \arrow[r, equal, "f(p)"] \arrow[d, equal, "H(x)"'] & f(y) \arrow[d, equal, "H(y)"] \\
g(x) \arrow[r, equal, "g(p)"'] & g(y)
\end{tikzcd}$$

proof of Lemma 2.4.3.

By induction, we may assume p is refl_x.Then trivially we observe that

$$\begin{aligned}
H(x) \cdot g(\mathsf{refl}_x) = f(\mathsf{refl}_x) \cdot H(x) \;&:\equiv H(x) \cdot \mathsf{ap}_g\ (\mathsf{refl}_x) = \mathsf{ap}_f\ (\mathsf{refl}_x) \cdot H(x) \\
&:\equiv H(x) \cdot \mathsf{refl}_x = \mathsf{refl}_x \cdot H(x) \\
&:\equiv H(x) = H(x)
\end{aligned}$$

which is inhabited by $\mathsf{refl}_{H(x)}$.

Quasi Inverse

Definitions

For a function $f : A \to B$ a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \to A$ and homotopies $\alpha : f \circ g \sim \mathsf{id}_B$ and $\beta : g \circ f \sim \mathsf{id}_A$.

Quasi Inverse

Definitions

For a function $f : A \to B$ a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \to A$ and homotopies $\alpha : f \circ g \sim \mathsf{id}_B$ and $\beta : g \circ f \sim \mathsf{id}_A$.
The **type of quasi-inverses** of f

$$\mathsf{QInv}\ (f) :\equiv \sum_{q:B\to A} ((f \circ g \sim \mathsf{id}_B) \times (g \circ f \sim \mathsf{id}_A)) \,.$$

Quasi Inverse

Definitions

For a function $f : A \to B$ a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \to A$ and homotopies $\alpha : f \circ g \sim \mathsf{id}_B$ and $\beta : g \circ f \sim \mathsf{id}_A$.
The **type of quasi-inverses** of f

$$\mathsf{QInv}\,(f) :\equiv \sum_{q:B\to A} \left((f \circ g \sim \mathsf{id}_B) \times (g \circ f \sim \mathsf{id}_A)\right).$$

We also define the types

$$\mathsf{LInv}\,(f) :\equiv \sum_{q:B\to A} (g \circ f \sim \mathsf{id}_A)$$

$$\mathsf{RInv}\,(f) :\equiv \sum_{q:B\to A} (f \circ g \sim \mathsf{id}_B)$$

of **left inverses** and **right inverses** to f , respectively.

Quasi Inverse

Definitions

For a function $f : A \to B$ a **quasi-inverse** of f is a triple (g, α, β) consisting of a function $g : B \to A$ and homotopies $\alpha : f \circ g \sim \mathsf{id}_B$ and $\beta : g \circ f \sim \mathsf{id}_A$.
The **type of quasi-inverses** of f

$$\mathsf{QInv}\ (f) :\equiv \sum_{q:B\to A} ((f \circ g \sim \mathsf{id}_B) \times (g \circ f \sim \mathsf{id}_A)).$$

We also define the types

$$\mathsf{LInv}\ (f) :\equiv \sum_{q:B\to A} (g \circ f \sim \mathsf{id}_A)$$

$$\mathsf{RInv}\ (f) :\equiv \sum_{q:B\to A} (f \circ g \sim \mathsf{id}_B)$$

of **left inverses** and **right inverses** to f , respectively.
We call f **left invertible** if $\mathsf{LInv}\ (f)$ is inhabited, and similarly **right invertible** if $\mathsf{RInv}\ (f)$ is inhabited.

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Thus we need something stronger. We want equivalence ($\mathsf{IsEquiv}(f)$) to have the following properties:

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Thus we need something stronger. We want equivalence ($\mathsf{IsEquiv}(f)$) to have the following properties:

1. $\mathsf{QInv}\ (f) \to \mathsf{IsEquiv}(f)$

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Thus we need something stronger. We want equivalence ($\mathsf{IsEquiv}(f)$) to have the following properties:

1. $\mathsf{QInv}\ (f) \to \mathsf{IsEquiv}(f)$
2. $\mathsf{IsEquiv}(f) \to \mathsf{QInv}\ (f)$

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Thus we need something stronger. We want equivalence ($\mathsf{IsEquiv}(f)$) to have the following properties:

1. $\mathsf{QInv}\ (f) \to \mathsf{IsEquiv}(f)$
2. $\mathsf{IsEquiv}(f) \to \mathsf{QInv}\ (f)$
3. $\mathsf{IsEquiv}(f)$ is a mere proposition.

Quasi Inverse vs Equivalence

Theorem (Theorem 4.1.3.)

Quasi Inverse is not a mere proposition.

Thus we need something stronger. We want equivalence ($\mathsf{IsEquiv}(f)$) to have the following properties:

1. $\mathsf{QInv}\ (f) \to \mathsf{IsEquiv}(f)$
2. $\mathsf{IsEquiv}(f) \to \mathsf{QInv}\ (f)$
3. $\mathsf{IsEquiv}(f)$ is a mere proposition.

We will firstly use our well known definition of equivalence:

$$\mathsf{IsEquiv}(f) :\equiv \mathsf{LInv}\ (f) \times \mathsf{RInv}\ (f)$$

Exercise (Exercise 2.10.)

Prove that Σ-types are "associative", in that for any $A : \mathcal{U}$ and families $B : A \to \mathcal{U}$ and $C : \sum_{(x:A)} B(x) \to \mathcal{U}$, we have

$$\left(\sum_{x:A} \sum_{y:B(x)} C(\mathsf{pair}(x, y)) \right) \simeq \left(\sum_{p:\sum_{(x:A)} B(x)} C(p) \right).$$

Exercise (Exercise 2.10.)

Prove that Σ-types are "associative", in that for any $A : \mathcal{U}$ and families $B : A \to \mathcal{U}$ and $C : \sum_{(x:A)} B(x) \to \mathcal{U}$, we have

$$\left(\sum_{x:A} \sum_{y:B(x)} C(\mathsf{pair}(x,y))\right) \simeq \left(\sum_{p:\sum_{(x:A)} B(x)} C(p)\right).$$

hint

Exercise (Exercise 2.10.)

Prove that Σ-types are "associative", in that for any $A : \mathcal{U}$ and families $B : A \to \mathcal{U}$ and $C : \sum_{(x:A)} B(x) \to \mathcal{U}$, we have

$$\left(\sum_{x:A} \sum_{y:B(x)} C(\mathsf{pair}(x,y))\right) \simeq \left(\sum_{p:\sum_{(x:A)} B(x)} C(p)\right).$$

hint

By induction for Σ-types

$$f :\equiv \mathsf{pair}\left(a, \mathsf{pair}\left(b_a, c_{\mathsf{pair}(a,b_a)}\right)\right) \mapsto \mathsf{pair}\left(\mathsf{pair}(a, b_a), c_{\mathsf{pair}(a,b_a)}\right)$$

$$g :\equiv \mathsf{pair}(u, c_u) \mapsto \mathsf{pair}(\mathsf{pr}_1(u), \mathsf{pair}(\mathsf{pr}_2(u), c_u))$$

Univalence

- Nov. 1853; George Boole

Univalence

- Nov. 1853; George Boole

  If instead of the proposition, "The sun shines," we say, "It is true that the sun shines," we then speak not directly of things, but of a proposition concerning things, viz., of the proposition, "The sun shines." And, therefore, the proposition in which we thus speak is a secondary one. Every primary proposition may thus give rise to a secondary proposition, viz., to that secondary proposition which asserts its truth, or declares its falsehood.

  An Investigation of the Laws of Thought, George Boole

Univalence

- Nov. 1853; George Boole

  If instead of the proposition, "The sun shines," we say, "It is true that the sun shines," we then speak not directly of things, but of a proposition concerning things, viz., of the proposition, "The sun shines." And, therefore, the proposition in which we thus speak is a secondary one. Every primary proposition may thus give rise to a secondary proposition, viz., to that secondary proposition which asserts its truth, or declares its falsehood.

  An Investigation of the Laws of Thought, George Boole

- 1935; Alfred Tarski; **Convention T**:

$$(''P'' = ''\mathsf{true}'') \simeq (P \simeq \mathsf{true})$$

Univalence

- Nov. 1853; George Boole

  If instead of the proposition, "The sun shines," we say, "It is true that the sun shines," we then speak not directly of things, but of a proposition concerning things, viz., of the proposition, "The sun shines." And, therefore, the proposition in which we thus speak is a secondary one. Every primary proposition may thus give rise to a secondary proposition, viz., to that secondary proposition which asserts its truth, or declares its falsehood.

  An Investigation of the Laws of Thought, George Boole

- 1935; Alfred Tarski; **Convention T**:

$$("P" = "\mathsf{true}") \simeq (P \simeq \mathsf{true})$$

"It is snowing" is a true sentence if and only if it is snowing

The Concept of Truth in Formalized Languages, Alfred Tarski

Univalence

- 1940; Alonzo Church (*A Formulation of the Simple Theory of Types*);

Univalence

- 1940; Alonzo Church (*A Formulation of the Simple Theory of Types*); **Propositional Extensionality**:

$$(P = Q) \simeq (P \simeq Q)\,,$$

  where P, Q are propositions.

Univalence

- 1940; Alonzo Church (*A Formulation of the Simple Theory of Types*); **Propositional Extensionality**:

  $$(P = Q) \simeq (P \simeq Q)\,,$$

  where P, Q are propositions.

- 1998; Martin Hofmann and Thomas Streicher [2]; Uniqueness of Identity Proofs (UIP) is not inhabited,

Univalence

- 1940; Alonzo Church (*A Formulation of the Simple Theory of Types*); **Propositional Extensionality**:

  $$(P = Q) \simeq (P \simeq Q)\,,$$

  where P, Q are propositions.

- 1998; Martin Hofmann and Thomas Streicher [2]; Uniqueness of Identity Proofs (UIP) is not inhabited, where $\mathsf{UIA}(A)$ stands for

  If a_1, a_2 are objects of type A then for any proofs p and q of the proposition "a_1 equals a_2" there is another proof establishing the equality of p and q.

Univalence

- 2006 - 2009; Vladimir Voevodsky **Univalence**

Univalence (aka UA)

For any $A, B : \mathcal{U}$, the function

$$\mathsf{idtoeqv} \colon (A =_{\mathcal{U}} B) \to (A \simeq B)$$

is an equivalence.

Univalence

- 2006 - 2009; Vladimir Voevodsky **Univalence**

Univalence (aka UA)

For any $A, B : \mathcal{U}$, the function

$$\mathsf{idtoeqv}\colon (A =_{\mathcal{U}} B) \to (A \simeq B)$$

is an equivalence.
In particular, therefore, we have

$$(A =_{\mathcal{U}} B) \simeq (A \simeq B)$$

Function Extensionality

What other kinds of extensionality implied by UA?

Function Extensionality

What other kinds of extensionality implied by UA?

Function Extensionality (aka FunExt)

For any $A, B : \mathcal{U}$ types and functions $f, g : A \to B$ the function

$$\mathsf{happly} \colon (f = g) \to \prod_{x:A} (f(x) =_B g(x))$$

is an equivalence.

Function Extensionality

What other kinds of extensionality implied by UA?

Function Extensionality (aka FunExt)

For any $A, B : \mathcal{U}$ types and functions $f, g : A \to B$ the function

$$\mathsf{happly}\colon (f = g) \to \prod_{x:A} (f(x) =_B g(x))$$

is an equivalence.
In particular happly has a quasi-inverse

$$\mathsf{funext}\colon \prod_{x:A} (f(x) =_B g(x)) \to (f = g).$$

Function Extensionality

What other kinds of extensionality implied by UA?

Function Extensionality (aka FunExt)

For any $A, B : \mathcal{U}$ types and functions $f, g : A \to B$ the function

$$\mathsf{happly} : (f = g) \to \prod_{x:A} (f(x) =_B g(x))$$

is an equivalence.
In particular happly has a quasi-inverse

$$\mathsf{funext} : \prod_{x:A} (f(x) =_B g(x)) \to (f = g).$$

Naive functional extensionality:
If functions take equal values, then they are equal.

Our Goal!

We want to show that

UA implies FunExt

Mere Propositions

Definition

A type P is a mere proposition if for all $x, y : P$ we have $x =_P y$.

Mere Propositions

Definition

A type P is a mere proposition if for all $x, y : P$ we have $x =_P y$. Specifically, for any $P : \mathcal{U}$, the type $\mathsf{IsProp}(P)$ is defined to be

$$\mathsf{IsProp}\,(P)\;:\equiv \prod_{x,y:P} (x =_P y)\,.$$

Mere Propositions

Definition

A type P is a mere proposition if for all $x, y : P$ we have $x =_P y$. Specifically, for any $P : \mathcal{U}$, the type $\mathsf{IsProp}(P)$ is defined to be

$$\mathsf{IsProp}(P) :\equiv \prod_{x,y:P} (x =_P y) .$$

Lemma (Lemma 3.3.3 / Λήμμα 45)

If P and Q are mere propositions such that $P \to Q$ and $Q \to P$, then $P \simeq Q$.

Contractability

Definition

A type A is **contractible**, or a **singleton**, if there is $a : A$, called the **center of contraction**, such that $a = x$ for all $x : A$. We denote the specified path $a = x$ by contr_x.
In other words, the type $\mathsf{IsContr}(A)$ is defined to be

$$\mathsf{IsContr}(A) :\equiv \sum_{a:A} \prod_{x:A} (a = x)\,.$$

Contractability

Definition

A type A is **contractible**, or a **singleton**, if there is $a : A$, called the **center of contraction**, such that $a = x$ for all $x : A$. We denote the specified path $a = x$ by contr_x.
In other words, the type $\mathsf{IsContr}(A)$ is defined to be

$$\mathsf{IsContr}(A) :\equiv \sum_{a:A} \prod_{x:A} (a = x).$$

Lemma (Lemma 3.11.8.)

For any A and any $a : A$, the type $\sum_{(x:A)} (a = x)$ is contractible.

Contractability

Definition

A type A is **contractible**, or a **singleton**, if there is $a : A$, called the **center of contraction**, such that $a = x$ for all $x : A$. We denote the specified path $a = x$ by contr_x.
In other words, the type $\mathsf{IsContr}(A)$ is defined to be

$$\mathsf{IsContr}(A) :\equiv \sum_{a:A} \prod_{x:A} (a = x).$$

Lemma (Lemma 3.11.8.)

For any A and any $a : A$, the type $\sum_{(x:A)} (a = x)$ is contractible.

Lemma (Lemma 3.11.9.)

Let $P : A \to \mathcal{U}$ be a type family.

1. *If each $P(x)$ is contractible, then $\sum_{(x:A)} P(x)$ is equivalent to A.*
2. *If A is contractible with center a, then $\sum_{(x:A)} P(x)$ is equivalent to $P(a)$.*

proof of Lemma 3.11.8.

We choose as center of the contraction the point pair (a, refl_a).

proof of Lemma 3.11.8.

We choose as center of the contraction the point $\mathsf{pair}(a, \mathsf{refl}_a)$. Now suppose $\mathsf{pair}(x, p) : \sum_{(x:A)} (a = x)$;

proof of Lemma 3.11.8.

We choose as center of the contraction the point $\mathsf{pair}(a, \mathsf{refl}_a)$. Now suppose $\mathsf{pair}(x, p) : \sum_{(x:A)} (a = x)$; we must show $\mathsf{pair}(a, \mathsf{refl}_a) = \mathsf{pair}(x, p)$.

proof of Lemma 3.11.8.

We choose as center of the contraction the point $\mathsf{pair}(a, \mathsf{refl}_a)$. Now suppose $\mathsf{pair}(x, p) : \sum_{(x:A)} (a = x)$; we must show $\mathsf{pair}(a, \mathsf{refl}_a) = \mathsf{pair}(x, p)$.

By the characterization of paths in Σ-types (Theorem 2.7.2. / Θεώρημα 32), we know that for any $w, w' : \sum_{(x:A)} (a = x)$, there is an equivalence

$$\left(w = w'\right) \simeq \sum_{(q:\mathsf{pr}_1(w) = \mathsf{pr}_1(w'))} \mathsf{transport}^{(a=-)}\left(q, \mathsf{pr}_2(w)\right) = \mathsf{pr}_2\left(w'\right).$$

proof of Lemma 3.11.8.

We choose as center of the contraction the point $\mathsf{pair}\,(a, \mathsf{refl}_a)$.
Now suppose $\mathsf{pair}\,(x, p) : \sum_{(x:A)} (a = x)$; we must show
$\mathsf{pair}\,(a, \mathsf{refl}_a) = \mathsf{pair}\,(x, p)$.

By the characterization of paths in Σ-types (Theorem 2.7.2. / Θεώρημα 32), we know that for any $w, w' : \sum_{(x:A)} (a = x)$, there is an equivalence

$$\left(w = w'\right) \simeq \sum_{(q:\mathsf{pr}_1(w) = \mathsf{pr}_1(w'))} \mathsf{transport}^{(a=-)}\,(q, \mathsf{pr}_2\,(w)) = \mathsf{pr}_2\,\left(w'\right).$$

Thus it suffices to exhibit $q : a = x$ such that
$\mathsf{transport}^{(a=-)}\,(q, \mathsf{refl}_a) = p$.

proof of Lemma 3.11.8.

We choose as center of the contraction the point $\mathsf{pair}(a, \mathsf{refl}_a)$.
Now suppose $\mathsf{pair}(x, p) : \sum_{(x:A)} (a = x)$; we must show $\mathsf{pair}(a, \mathsf{refl}_a) = \mathsf{pair}(x, p)$.

By the characterization of paths in Σ-types (Theorem 2.7.2. / Θεώρημα 32), we know that for any $w, w' : \sum_{(x:A)} (a = x)$, there is an equivalence

$$(w = w') \simeq \sum_{(q:\mathsf{pr}_1(w) = \mathsf{pr}_1(w'))} \mathsf{transport}^{(a=-)}(q, \mathsf{pr}_2(w)) = \mathsf{pr}_2(w').$$

Thus it suffices to exhibit $q : a = x$ such that $\mathsf{transport}^{(a=-)}(q, \mathsf{refl}_a) = p$.
But we can take $q :\equiv p$ in which case

$$\begin{aligned} \mathsf{transport}^{(a=-)}(q, \mathsf{refl}_a) &= p \cdot \mathsf{refl}_a && \text{L.2.11.2. / Λήμμα 24} \\ &= p && \text{L.2.11.4. / Λήμμα 15} \end{aligned}$$

Retract

If A is equivalent to B and A is contractible, then so is B.

Retract

If A is equivalent to B and A is contractible, then so is B.
More generally, it suffices for B to be a *retract* of A.

Retract

If A is equivalent to B and A is contractible, then so is B.
More generally, it suffices for B to be a *retract* of A.

Definition

A **retraction** is a function $r : A \to B$ such that there exists a function $s : B \to A$, called its **section**, and a homotopy $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

Retract

If A is equivalent to B and A is contractible, then so is B.
More generally, it suffices for B to be a *retract* of A.

Definition

A **retraction** is a function $r : A \to B$ such that there exists a function $s : B \to A$, called its **section**, and a homotopy $\epsilon : \prod_{(y:B)} (r(s(y)) = y) \equiv r \circ s \sim \mathsf{id}_A$;

Retract

If A is equivalent to B and A is contractible, then so is B.
More generally, it suffices for B to be a *retract* of A.

Definition

A **retraction** is a function $r : A \to B$ such that there exists a function $s : B \to A$, called its **section**, and a homotopy $\epsilon : \prod_{(y:B)} (r(s(y)) = y) \equiv r \circ s \sim \mathsf{id}_A$;
then we say that B is a **retract** of A.

Retract

If A is equivalent to B and A is contractible, then so is B.
More generally, it suffices for B to be a *retract* of A.

Definition

A **retraction** is a function $r : A \to B$ such that there exists a function $s : B \to A$, called its **section**, and a homotopy
$\epsilon : \prod_{(y:B)} (r\,(s\,(y)) = y) \equiv r \circ s \sim \mathsf{id}_A$;
then we say that B is a **retract** of A.

Lemma (Lemma 3.11.7.)

If B is a retract of A, and A is contractible, then so is B.

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction.

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r\,(s\,(y)) = y)$

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

We claim that $b_0 :\equiv r(a_0) : B$ is a center of contraction for B.

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

We claim that $b_0 :\equiv r(a_0) : B$ is a center of contraction for B.

Let $b : B$;

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

We claim that $b_0 :\equiv r(a_0) : B$ is a center of contraction for B.

Let $b : B$; we need a path $p : b_0 = b$.

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r\,(s\,(y)) = y)$

We claim that $b_0 :\equiv r\,(a_0) : B$ is a center of contraction for B.

Let $b : B$; we need a path $p : b_0 = b$.
But we have $\epsilon\,(b) : r \circ s\,(b) = b$ and $\mathsf{contr}_{s(b)} : a_0 = s\,(b)$,

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

We claim that $b_0 :\equiv r(a_0) : B$ is a center of contraction for B.

Let $b : B$; we need a path $p : b_0 = b$.
But we have $\epsilon(b) : r \circ s(b) = b$ and $\mathsf{contr}_{s(b)} : a_0 = s(b)$, so by composition

$$r\left(\mathsf{contr}_{s(b)}\right) :\equiv \mathsf{ap}_r\left(\mathsf{contr}_{s(b)}\right) : r(a_0) = r \circ s(b)$$

proof of Lemma 3.11.7.

Let $a_0 : A$ be the center of contraction. Let also,

- $r : A \to B$ the retraction
- $s : B \to A$ the section
- $\epsilon : \prod_{(y:B)} (r(s(y)) = y)$

We claim that $b_0 :\equiv r(a_0) : B$ is a center of contraction for B.

Let $b : B$; we need a path $p : b_0 = b$.
But we have $\epsilon(b) : r \circ s(b) = b$ and $\mathsf{contr}_{s(b)} : a_0 = s(b)$, so by composition

$$r\left(\mathsf{contr}_{s(b)}\right) :\equiv \mathsf{ap}_r\left(\mathsf{contr}_{s(b)}\right) : r(a_0) = r \circ s(b)$$

thus

$$r\left(\mathsf{contr}_{s(b)}\right) \cdot \epsilon(b) : b_0 = b.$$

We conclude that B is contractible with center of contraction b_0.

∎

Contractible fibers

Definitions

The **fiber** (ἴνα) of a map $f : A \to B$ over a point $y : B$ is

$$\mathsf{fib}_f(y) :\equiv \sum_{x:A} (f(x) = y).$$

Contractible fibers

Definitions

The **fiber** (ἶνα) of a map $f : A \to B$ over a point $y : B$ is

$$\mathsf{fib}_f\,(y) :\equiv \sum_{x:A} (f\,(x) = y)\,.$$

In homotopy theory, this is what would be called the *homotopy fiber* of f.

Contractible fibers

Definitions

The **fiber** (ίνα) of a map $f : A \to B$ over a point $y : B$ is

$$\mathsf{fib}_f(y) :\equiv \sum_{x:A} (f(x) = y).$$

In homotopy theory, this is what would be called the *homotopy fiber* of f.

A map $f : A \to B$ is **contractible** if for all $y : B$, the fiber $\mathsf{fib}_f(y)$ is contractible.

Contractible fibers

Definitions

The **fiber** (ἶνα) of a map $f : A \to B$ over a point $y : B$ is

$$\mathsf{fib}_f\,(y) :\equiv \sum_{x:A} (f\,(x) = y)\,.$$

In homotopy theory, this is what would be called the *homotopy fiber* of f.

A map $f : A \to B$ is **contractible** if for all $y : B$, the fiber $\mathsf{fib}_f\,(y)$ is contractible.
Thus the type $\mathsf{IsContr}\,(f)$ is defined to be

$$\mathsf{IsContr}\,(f) :\equiv \prod_{y:B} \mathsf{IsContr}\left(\mathsf{fib}_f\,(y)\right).$$

A Useful Lemma

We are going to need the following lemma.

A Useful Lemma

We are going to need the following lemma.

Lemma (Lemma 4.8.1.)

For any type family $B : A \to \mathcal{U}$, the fiber of $\mathsf{pr}_1 \colon \sum_{(x:A)} B\,(x) \to A$ over $a : A$ is equivalent to $B(a)$:

$$\mathsf{fib}_{\mathsf{pr}_1}\,(a) \simeq B\,(a)\,.$$

proof of Lemma 4.8.1. We have

$$\mathsf{fib}_{\mathsf{pr}_1}(a) :\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a)$$

proof of Lemma 4.8.1. We have

$$
\begin{aligned}
\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a) \\
&\simeq \sum_{x:A} \sum_{b:B(x)} (x = a) && \text{Ex. 2.10}
\end{aligned}
$$

proof of Lemma 4.8.1. We have

$$
\begin{aligned}
\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a) \\
&\simeq \sum_{x:A} \sum_{b:B(x)} (x = a) && \text{Ex. 2.10} \\
&\simeq \sum_{x:A} \sum_{p:x=a} B(x) && (*)
\end{aligned}
$$

proof of Lemma 4.8.1. We have

$$
\begin{aligned}
\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a) \\
&\simeq \sum_{x:A} \sum_{b:B(x)} (x = a) && \text{Ex. 2.10} \\
&\simeq \sum_{x:A} \sum_{p:x=a} B(x) && (*) \\
&\simeq B(a) && (**)
\end{aligned}
$$

proof of Lemma 4.8.1. We have

$$\begin{aligned}
\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a) \\
&\simeq \sum_{x:A} \sum_{b:B(x)} (x = a) && \text{Ex. 2.10} \\
&\simeq \sum_{x:A} \sum_{p:x=a} B(x) && (*) \\
&\simeq B(a) && (**)
\end{aligned}$$

$$(*) \qquad \begin{aligned}
f &:\equiv \mathsf{pair}(a, \mathsf{pair}(b_a, \mathsf{refl}_a)) \mapsto \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \\
g &:\equiv \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \mapsto \mathsf{pair}(a, \mathsf{pair}(b_a, \mathsf{refl}_a))
\end{aligned}$$

proof of Lemma 4.8.1. We have

$$\begin{aligned}
\mathsf{fib}_{\mathsf{pr}_1}(a) &:\equiv \sum_{u:\sum_{(x:A)} B(x)} (\mathsf{pr}_1(u) = a) \\
&\simeq \sum_{x:A} \sum_{b:B(x)} (x = a) && \text{Ex. 2.10} \\
&\simeq \sum_{x:A} \sum_{p:x=a} B(x) && (*) \\
&\simeq B(a) && (**)
\end{aligned}$$

$$(*) \qquad \begin{aligned} f &:\equiv \mathsf{pair}(a, \mathsf{pair}(b_a, \mathsf{refl}_a)) \mapsto \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \\ g &:\equiv \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \mapsto \mathsf{pair}(a, \mathsf{pair}(b_a, \mathsf{refl}_a)) \end{aligned}$$

$$(**) \qquad \begin{aligned} f &:\equiv \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \mapsto b_a \\ g &:\equiv b_a \mapsto \mathsf{pair}(a, \mathsf{pair}(\mathsf{refl}_a, b_a)) \end{aligned}$$

□

Definition

A function $g : A \to B$ is said to be a **retract** of a function $f : X \to Y$ if there is a diagram

$$\begin{tikzcd} A \arrow[r, "s"] \arrow[d, "g"'] & X \arrow[r, "r"] \arrow[d, "f"'] & A \arrow[d, "g"] \\ B \arrow[r, "s'"'] & Y \arrow[r, "r'"'] & B \end{tikzcd}$$

for which there are

1. a homotopy $R : r \circ s \sim \mathrm{id}_A$
2. a homotopy $R' : r' \circ s' \sim \mathrm{id}_B$
3. a homotopy $L : f \circ s \sim s' \circ g$
4. a homotopy $K : g \circ r \sim r' \circ f$
5. for every $a : A$, a path $H(a)$ witnessing the commutativity of the square

$$\begin{tikzcd} g(r(s(a))) \arrow[r, equal, "K(s(a))"] \arrow[d, equal, "g(R(a))"'] & r'(f(s(a))) \arrow[d, equal, "r'(L(a))"] \\ g(a) \arrow[r, equal, "(R'(g(a)))^{-1}"'] & r'(s'(g(a))) \end{tikzcd}$$

Equivalences

We have the following 3 approaches of the notion of equivalence.

Definitions

1. **Half Adjoint Equivalence** (definition used in HoTT)
   A function $f : A \to B$ is a half adjoint equivalence if there are $g : B \to A$ and homotopies $\eta : g \circ f \sim \mathrm{id}_A$ and $\epsilon : f \circ g \sim \mathrm{id}_B$ such that there exists a homotopy
   $$\tau : \prod_{x:A} (f(\eta(x)) = \epsilon(f(x))).$$

Equivalences

We have the following 3 approaches of the notion of equivalence.

Definitions

1. **Half Adjoint Equivalence** (definition used in HoTT)
   A function $f : A \to B$ is a half adjoint equivalence if there are $g : B \to A$ and homotopies $\eta : g \circ f \sim \mathsf{id}_A$ and $\epsilon : f \circ g \sim \mathsf{id}_B$ such that there exists a homotopy
   $$\tau : \prod_{x:A} \left(f\left(\eta\left(x\right)\right) = \epsilon\left(f\left(x\right)\right)\right).$$
   Thus we have a type $\mathsf{ishae}(f)$, defined to be
   $$\mathsf{ishae}(f) :\equiv \sum_{(g:B\to A)} \sum_{(\eta:g\circ f\sim\mathsf{id}_A)} \sum_{(\epsilon:f\circ g\sim\mathsf{id}_B)} \prod_{x:A} \left(f\left(\eta\left(x\right)\right) = \epsilon\left(f\left(x\right)\right)\right)$$

Equivalences

We have the following 3 approaches of the notion of equivalence.

Definitions

1. **Half Adjoint Equivalence** (definition used in HoTT)
   A function $f : A \to B$ is a half adjoint equivalence if there are $g : B \to A$ and homotopies $\eta : g \circ f \sim \mathsf{id}_A$ and $\epsilon : f \circ g \sim \mathsf{id}_B$ such that there exists a homotopy
   $$\tau : \prod_{x:A} (f(\eta(x)) = \epsilon(f(x))).$$
   Thus we have a type $\mathsf{ishae}(f)$, defined to be
   $$\mathsf{ishae}(f) :\equiv \sum_{(g:B \to A)} \sum_{(\eta : g \circ f \sim \mathsf{id}_A)} \sum_{(\epsilon : f \circ g \sim \mathsf{id}_B)} \prod_{x:A} (f(\eta(x)) = \epsilon(f(x)))$$

2. **Bi-invertible Map** (the well known definition)

Equivalences

We have the following 3 approaches of the notion of equivalence.

Definitions

1. **Half Adjoint Equivalence** (definition used in HoTT)
   A function $f : A \to B$ is a half adjoint equivalence if there are $g : B \to A$ and homotopies $\eta : g \circ f \sim \mathsf{id}_A$ and $\epsilon : f \circ g \sim \mathsf{id}_B$ such that there exists a homotopy
   $$\tau : \prod_{x:A} \left(f\left(\eta\left(x\right)\right) = \epsilon\left(f\left(x\right)\right)\right).$$
   Thus we have a type $\mathsf{ishae}(f)$, defined to be
   $$\mathsf{ishae}(f) :\equiv \sum_{(g:B\to A)} \sum_{(\eta:g\circ f\sim \mathsf{id}_A)} \sum_{(\epsilon:f\circ g\sim \mathsf{id}_B)} \prod_{x:A} \left(f\left(\eta\left(x\right)\right) = \epsilon\left(f\left(x\right)\right)\right)$$
2. **Bi-invertible Map** (the well known definition)
3. **Contractible Functions** (the one used by Voevodsky)

We will mainly use the 1st definition of equivalence.

We will mainly use the 1st definition of equivalence.

But are those definitions equivalences?

We will mainly use the 1st definition of equivalence.

But are those definitions equivalences?

Theorem (Theorem 4.2.3.)

For any $f : A \to B$ *we have* $\mathsf{QInv}\ (f) \to \mathsf{IsHaE}(f)$.

We will mainly use the 1st definition of equivalence.

But are those definitions equivalences?

Theorem (Theorem 4.2.3.)

For any $f : A \to B$ *we have* $\mathsf{QInv}\ (f) \to \mathsf{IsHaE}(f)$.

The other direction is trivial. (Why?)

proof of the Theorem 4.2.3.

Let $(g, \eta, \epsilon) : \mathsf{QInv}\ (f)$.

proof of the Theorem 4.2.3.

Let (g, η, ϵ) :QInv (f).

We want to to provide a quadruple $(g', \eta', \epsilon', \tau)$: $\mathsf{IsHaE}(f)$.

proof of the Theorem 4.2.3.

Let (g, η, ϵ) :$\mathsf{QInv}\ (f)$.

We want to to provide a quadruple $(g', \eta', \epsilon', \tau)$: $\mathsf{IsHaE}(f)$.

We define

- $g' :\equiv g$
- $\eta' :\equiv \eta$
- $\epsilon' :\equiv \epsilon\,(f\,(g\,(b)))^{-1} \cdot f\,(\eta\,(g\,(b))) \cdot \epsilon\,(b)$

proof of the Theorem 4.2.3.

Let (g, η, ϵ) :QInv (f).

We want to to provide a quadruple $(g', \eta', \epsilon', \tau)$: $\mathsf{IsHaE}(f)$.

We define

- $g' :\equiv g$
- $\eta' :\equiv \eta$
- $\epsilon' :\equiv \epsilon\left(f\left(g\left(b\right)\right)\right)^{-1} \cdot f\left(\eta\left(g\left(b\right)\right)\right) \cdot \epsilon\left(b\right) \quad ?!$

proof of the Theorem 4.2.3.

Let (g, η, ϵ) :QInv (f).

We want to to provide a quadruple $\left(g', \eta', \epsilon', \tau\right)$: $\mathsf{IsHaE}(f)$.

We define

- $g' :\equiv g$
- $\eta' :\equiv \eta$
- $\epsilon' :\equiv \epsilon\left(f\left(g\left(b\right)\right)\right)^{-1} \cdot f\left(\eta\left(g\left(b\right)\right)\right) \cdot \epsilon\left(b\right)$?!

Let us brake ϵ' into pieces!

- $\epsilon\left(f\left(g\left(b\right)\right)\right)^{-1} : f \circ g\left(b\right) = f \circ g\left(f\left(g\left(b\right)\right)\right)$
- $f\left(\eta\left(g\left(b\right)\right)\right) : f \circ g\left(f\left(g\left(b\right)\right)\right) = f \circ g\left(b\right)$
- $\epsilon\left(b\right) : f \circ g\left(b\right) = b$

proof of the Theorem 4.2.3.

Let (g, η, ϵ) :QInv (f).

We want to to provide a quadruple $\left(g', \eta', \epsilon', \tau\right)$: IsHaE(f).

We define

- $g' :\equiv g$
- $\eta' :\equiv \eta$
- $\epsilon' :\equiv \epsilon\left(f\left(g\left(b\right)\right)\right)^{-1} \cdot f\left(\eta\left(g\left(b\right)\right)\right) \cdot \epsilon\left(b\right)$?!

Let us brake ϵ' into pieces!

- $\epsilon\left(f\left(g\left(b\right)\right)\right)^{-1} : f \circ g\left(b\right) = f \circ g\left(f\left(g\left(b\right)\right)\right)$
- $f\left(\eta\left(g\left(b\right)\right)\right) : f \circ g\left(f\left(g\left(b\right)\right)\right) = f \circ g\left(b\right)$
- $\epsilon\left(b\right) : f \circ g\left(b\right) = b$

Thus $\epsilon' : f \circ g\left(b\right) = b$ as wanted.

We need to find τ, s.t.

$$\tau\left(a\right) : f\left(\eta\left(a\right)\right) = \epsilon'\left(a\right).$$

proof of the theorem (Cont'd)

From Lemma 2.4.3. we can easilly observe that

$$\eta\left(g \circ f\left(a\right)\right) = g \circ f\left(\eta\left(a\right)\right). \tag{1}$$

proof of the theorem (Cont'd)

From Lemma 2.4.3. we can easilly observe that

$$\eta\left(g \circ f\left(a\right)\right) = g \circ f\left(\eta\left(a\right)\right). \tag{1}$$

Therefore,

$$\begin{aligned} f\left(\eta\left(g \circ f\left(a\right)\right)\right) \cdot \epsilon\left(f\left(a\right)\right) &= f \circ g\left(f\left(\eta\left(a\right)\right)\right) \cdot \epsilon\left(f\left(a\right)\right) && 1 \\ &= \epsilon\left(f\left(g \circ f\left(a\right)\right)\right) \cdot f\left(\eta\left(a\right)\right) && \text{Lemma 2.4.3.} \end{aligned}$$

proof of the theorem (Cont'd)

From Lemma 2.4.3. we can easilly observe that

$$\eta\left(g \circ f\left(a\right)\right) = g \circ f\left(\eta\left(a\right)\right). \tag{1}$$

Therefore,

$$\begin{aligned} f\left(\eta\left(g \circ f\left(a\right)\right)\right) \cdot \epsilon\left(f\left(a\right)\right) &= f \circ g\left(f\left(\eta\left(a\right)\right)\right) \cdot \epsilon\left(f\left(a\right)\right) && 1 \\ &= \epsilon\left(f\left(g \circ f\left(a\right)\right)\right) \cdot f\left(\eta\left(a\right)\right) && \text{Lemma 2.4.3.} \end{aligned}$$

where we used Lemma 2.4.3. as

- $f \leftarrow f \circ g$ and $g \leftarrow \mathrm{id}_A$
- $H \leftarrow \epsilon$
- $x \leftarrow f \circ g\left(f\left(a\right)\right)$ and $y \leftarrow f\left(a\right)$
- $p \leftarrow f\left(\eta\left(a\right)\right) : f \circ g\left(f\left(a\right)\right) = f\left(a\right)$

$\square$

Equivalence of Equivalences :)

Theorem (Theorem 4.2.13.)

For any $f : A \to B$*, the type* $\mathsf{IsHaE}(f)$ *is a mere proposition.*

Equivalence of Equivalences :)

Theorem (Theorem 4.2.13.)

For any $f : A \to B$*, the type* $\mathsf{IsHaE}\,(f)$ *is a mere proposition.*

Theorem (Corollary 4.3.3. & Theorem 4.4.5.)

All three types $\mathsf{IsHaE}, \mathsf{BiInv}$ *and* $\mathsf{IsContr}$ *are equivalent:*

$$\mathsf{IsHaE} \simeq \mathsf{BiInv} \simeq \mathsf{IsContr}$$

Equivalence of Equivalences :)

Theorem (Theorem 4.2.13.)

For any $f : A \to B$*, the type* $\mathsf{IsHaE}\,(f)$ *is a mere proposition.*

Theorem (Corollary 4.3.3. & Theorem 4.4.5.)

All three types $\mathsf{IsHaE}, \mathsf{BiInv}$ *and* $\mathsf{IsContr}$ *are equivalent:*

$$\mathsf{IsHaE} \simeq \mathsf{BiInv} \simeq \mathsf{IsContr}$$

Strategy of proof:
$\mathsf{BiInv}\,(f) \leftrightarrow \mathsf{IsHaE}\,(f)$ *and* $\mathsf{IsContr}\,(f) \leftrightarrow \mathsf{IsHaE}\,(f)$ *and*
$\mathsf{BiInv}\,(f)$, $\mathsf{IsContr}\,(f)$ *are mere propositions (Lemma 3.3.3 / Λήμμα 45)*

Total Space

Definitions

- Given two type families $P, Q : A \to \mathcal{U}$, we refer to a function $f : \prod_{(x:A)} (P(x) \to Q(x))$ as a **fiberwise map** or a **fiberwise transformation**.

Total Space

Definitions

- Given two type families $P, Q : A \to \mathcal{U}$, we refer to a function $f : \prod_{(x:A)} (P(x) \to Q(x))$ as a **fiberwise map** or a **fiberwise transformation**.
- Such a map induces a function on **total spaces**

$$\mathsf{total}(f) :\equiv \lambda w.\, \mathsf{pair}(\mathsf{pr}_1(w), f(\mathsf{pair}(\mathsf{pr}_1(w), \mathsf{pr}_2(w)))) : \sum_{x:A} P(x) \to \sum_{x:A} Q(x).$$

Total Space

Definitions

- Given two type families $P, Q : A \to \mathcal{U}$, we refer to a function $f : \prod_{(x:A)} (P(x) \to Q(x))$ as a **fiberwise map** or a **fiberwise transformation**.
- Such a map induces a function on **total spaces**

$$\mathsf{total}(f) :\equiv \lambda w.\, \mathsf{pair}(\mathsf{pr}_1(w), f(\mathsf{pair}(\mathsf{pr}_1(w), \mathsf{pr}_2(w)))) : \sum_{x:A} P(x) \to \sum_{x:A} Q(x).$$

- We say that a fiberwise map $f : \prod_{(x:A)} (P(x) \to Q(x))$ is a **fiberwise equivalence** if each $f(x) : P(x) \to Q(x)$ is an equivalence.

Total Space

Theorem (Theorem 4.7.6.)

Suppose that f is a fiberwise transformation between families P and Q over a type A and let $x : A$ and $v : Q(x)$. Then we have an equivalence

$$\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right) \simeq \mathsf{fib}_{f(x)}\left(v\right).$$

Total Space

Theorem (Theorem 4.7.6.)

Suppose that f is a fiberwise transformation between families P and Q over a type A and let $x : A$ and $v : Q(x)$. Then we have an equivalence

$$\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}(x,v)\right) \simeq \mathsf{fib}_{f(x)}(v).$$

Theorem (Theorem 4.7.7.)

Suppose that f is a fiberwise transformation between families P and Q over a type A. Then f is a fiberwise equivalence if and only if $\mathsf{total}(f)$ is an equivalence.

proof of the theorem 4.7.6.

We calculate:

$$\mathsf{fib}_{\mathsf{total}(f)}\,(\mathsf{pair}(x,v)) :\equiv \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}(\mathsf{pr}_1\,(w)\,, f\,(\mathsf{pair}(\mathsf{pr}_1\,(w)\,, \mathsf{pr}_2\,(w)))) = \quad \mathsf{pair}(x,v)$$

proof of the theorem 4.7.6.

We calculate:

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}(x,v)) :\equiv \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}(\mathsf{pr}_1\ (w), f\ (\mathsf{pair}(\mathsf{pr}_1\ (w), \mathsf{pr}_2\ (w)))) = \mathsf{pair}(x,v)$$

$$\simeq \sum_{a:A} \sum_{u:P(a)} \mathsf{pair}(a, f\ (\mathsf{pair}(a,u))) = \mathsf{pair}(x,v) \qquad \text{Ex.2.10.}$$

proof of the theorem 4.7.6.

We calculate:

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}(x,v)) :\equiv \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}(\mathsf{pr}_1\ (w), f\ (\mathsf{pair}(\mathsf{pr}_1\ (w), \mathsf{pr}_2\ (w)))) = \quad \mathsf{pair}(x,v)$$

$$\simeq \sum_{a:A} \sum_{u:P(a)} \mathsf{pair}(a, f\ (\mathsf{pair}(a,u))) = \mathsf{pair}(x,v) \qquad \text{Ex.2.10.}$$

$$\simeq \sum_{a:A} \sum_{u:P(a)} \sum_{p:a=x} \mathsf{transport}^{(a=-)}\ (p, f\ (\mathsf{pair}(a,u))) = v \qquad \Theta.\ 32.$$

proof of the theorem 4.7.6.

We calculate:

$$\mathsf{fib}_{\mathsf{total}(f)}\,(\mathsf{pair}\,(x,v)) :\equiv \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}\,(\mathsf{pr}_1\,(w)\,,f\,(\mathsf{pair}\,(\mathsf{pr}_1\,(w)\,,\mathsf{pr}_2\,(w)))) = \quad \mathsf{pair}\,(x,v)$$

$$\simeq \sum_{a:A}\sum_{u:P(a)} \mathsf{pair}\,(a,f\,(\mathsf{pair}\,(a,u))) = \mathsf{pair}\,(x,v) \qquad \text{Ex.2.10.}$$

$$\simeq \sum_{a:A}\sum_{u:P(a)}\sum_{p:a=x} \mathsf{transport}^{(a=-)}\,(p,f\,(\mathsf{pair}\,(a,u))) = v \qquad \Theta.\ 32.$$

$$\simeq \sum_{a:A}\sum_{p:a=x}\sum_{u:P(a)} \mathsf{transport}^{(a=-)}\,(p,f\,(\mathsf{pair}\,(a,u))) = v$$

proof of the theorem 4.7.6.

We calculate:

$$
\begin{aligned}
\mathsf{fib}_{\mathsf{total}(f)}\,(\mathsf{pair}(x,v)) :\equiv& \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}(\mathsf{pr}_1\,(w), f\,(\mathsf{pair}(\mathsf{pr}_1\,(w), \mathsf{pr}_2\,(w))))= \quad \mathsf{pair}(x,v) \\
\simeq& \sum_{a:A}\sum_{u:P(a)} \mathsf{pair}(a, f\,(\mathsf{pair}(a,u))) = \mathsf{pair}(x,v) && \text{Ex.2.10.} \\
\simeq& \sum_{a:A}\sum_{u:P(a)}\sum_{p:a=x} \mathsf{transport}^{(a=-)}\,(p, f\,(\mathsf{pair}(a,u))) = v && \Theta.\ 32. \\
\simeq& \sum_{a:A}\sum_{p:a=x}\sum_{u:P(a)} \mathsf{transport}^{(a=-)}\,(p, f\,(\mathsf{pair}(a,u))) = v \\
\simeq& \sum_{u:P(x)} f\,(\mathsf{pair}(x,u)) = v && (*)
\end{aligned}
$$

proof of the theorem 4.7.6.

We calculate:

$$
\begin{aligned}
\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}(x,v)) &:\equiv \sum_{w:\sum_{(x:A)} P(x)} \mathsf{pair}(\mathsf{pr}_1\ (w), f\ (\mathsf{pair}(\mathsf{pr}_1\ (w), \mathsf{pr}_2\ (w)))) = \quad \mathsf{pair}(x,v) \\
&\simeq \sum_{a:A} \sum_{u:P(a)} \mathsf{pair}(a, f\ (\mathsf{pair}(a,u))) = \mathsf{pair}(x,v) && \text{Ex.2.10.} \\
&\simeq \sum_{a:A} \sum_{u:P(a)} \sum_{p:a=x} \mathsf{transport}^{(a=-)}\ (p, f\ (\mathsf{pair}(a,u))) = v && \Theta.\ 32. \\
&\simeq \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\ (p, f\ (\mathsf{pair}(a,u))) = v \\
&\simeq \sum_{u:P(x)} f\ (\mathsf{pair}(x,u)) = v && (*) \\
&\equiv \mathsf{fib}_{f(x)}\ (v)
\end{aligned}
$$

proof of the theorem 4.7.6. (Cont'd)

$$(*) \quad \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v \simeq sum_{u:P(x)} f\left(\mathsf{pair}(x,u)\right) = v$$

By Lemma 3.11.8. $\sum_{(x:A)} (a = x)$ is contractible with center of contraction

(a, refl_a).

proof of the theorem 4.7.6. (Cont'd)

$$(*) \quad \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v \simeq sum_{u:P(x)} f\left(\mathsf{pair}(x,u)\right) = v$$

By Lemma 3.11.8. $\sum_{(x:A)} (a = x)$ is contractible with center of contraction

(a, refl_a).
Let us assume that $P : \sum_{(x:A)} (a = x) \to \mathcal{U}$, where

$$P\left(\mathsf{pair}\left(a,p\right)\right) :\equiv \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v.$$

proof of the theorem 4.7.6. (Cont'd)

$$(*) \quad \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}(p, f(\mathsf{pair}(a,u))) = v \simeq sum_{u:P(x)} f(\mathsf{pair}(x,u)) = v$$

By Lemma 3.11.8. $\sum_{(x:A)}(a = x)$ is contractible with center of contraction

(a, refl_a).
Let us assume that $P : \sum_{(x:A)}(a = x) \to \mathcal{U}$, where

$$P(\mathsf{pair}(a,p)) :\equiv \sum_{u:P(a)} \mathsf{transport}^{(a=-)}(p, f(\mathsf{pair}(a,u))) = v.$$

By Lemma 3.11.9. and Exercise 2.10. we have that

proof of the theorem 4.7.6. (Cont'd)

$$(*) \quad \sum_{a:A}\sum_{p:a=x}\sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v \simeq sum_{u:P(x)} f\left(\mathsf{pair}(x,u)\right) = v$$

By Lemma 3.11.8. $\sum_{(x:A)}(a = x)$ is contractible with center of contraction

(a, refl_a).
Let us assume that $P : \sum_{(x:A)}(a = x) \to \mathcal{U}$, where

$$P\left(\mathsf{pair}(a,p)\right) :\equiv \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v.$$

By Lemma 3.11.9. and Exercise 2.10. we have that

$$\sum_{a:A}\sum_{p:a=x}\sum_{u:P(a)} \mathsf{transport}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v \simeq \sum_{u:P(a)} \mathsf{transport}\left(\mathsf{refl}_a, f\left(\mathsf{pair}(a,u)\right)\right) = v$$

$$(*) \quad \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v \simeq sum_{u:P(x)} f\left(\mathsf{pair}(x,u)\right) = v$$

By Lemma 3.11.8. $\sum_{(x:A)} (a = x)$ is contractible with center of contraction

(a, refl_a).
Let us assume that $P : \sum_{(x:A)} (a = x) \to \mathcal{U}$, where

$$P\left(\mathsf{pair}(a,p)\right) :\equiv \sum_{u:P(a)} \mathsf{transport}^{(a=-)}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v.$$

By Lemma 3.11.9. and Exercise 2.10. we have that

$$\begin{aligned}\sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}\left(p, f\left(\mathsf{pair}(a,u)\right)\right) = v &\simeq \sum_{u:P(a)} \mathsf{transport}\left(\mathsf{refl}_a, f\left(\mathsf{pair}(a,u)\right)\right) = v \\ &\equiv \sum_{u:P(x)} f\left(\mathsf{pair}(x,u)\right) = v\end{aligned}$$

proof of the theorem 4.7.6. (Cont'd)

$$(*) \quad \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}^{(a=-)}(p, f(\mathsf{pair}(a,u))) = v \simeq sum_{u:P(x)} f(\mathsf{pair}(x,u)) = v$$

By Lemma 3.11.8. $\sum_{(x:A)} (a = x)$ is contractible with center of contraction

(a, refl_a).
Let us assume that $P : \sum_{(x:A)} (a = x) \to \mathcal{U}$, where

$$P(\mathsf{pair}(a,p)) :\equiv \sum_{u:P(a)} \mathsf{transport}^{(a=-)}(p, f(\mathsf{pair}(a,u))) = v.$$

By Lemma 3.11.9. and Exercise 2.10. we have that

$$\begin{aligned} \sum_{a:A} \sum_{p:a=x} \sum_{u:P(a)} \mathsf{transport}(p, f(\mathsf{pair}(a,u))) = v &\simeq \sum_{u:P(a)} \mathsf{transport}(\mathsf{refl}_a, f(\mathsf{pair}(a,u))) = v \\ &\equiv \sum_{u:P(x)} f(\mathsf{pair}(x,u)) = v \end{aligned}$$

$\square$

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\,(\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\,(v)\,.$$

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right) \simeq \mathsf{fib}_{f(x)}\left(v\right).$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right)$ is contractible iff $\mathsf{fib}_{f(x)}\left(v\right)$ is contractible.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right) \simeq \mathsf{fib}_{f(x)}\left(v\right).$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right)$ is contractible iff $\mathsf{fib}_{f(x)}\left(v\right)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\left(x\right) \to Q\left(x\right)$ is a fiberwise equivalence

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\,(x) \to Q\,(x)$ is a fiberwise equivalence
- Iff for all $x : A,\ f\,(x) : P\,(x) \to Q\,(x)$ is an equivalence.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\,(x) \to Q\,(x)$ is a fiberwise equivalence
- Iff for all $x : A$, $f\,(x) : P\,(x) \to Q\,(x)$ is an equivalence.
- Iff for all $x : A$, $f\,(x)$ is contractible.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P(x) \to Q(x)$ is a fiberwise equivalence
- Iff for all $x : A$, $f(x) : P(x) \to Q(x)$ is an equivalence.
- Iff for all $x : A$, $f(x)$ is contractible.
- Iff for all $x : A$ and for all $v \in Q(x)$, $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right) \simeq \mathsf{fib}_{f(x)}\left(v\right).$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\left(\mathsf{pair}\left(x, v\right)\right)$ is contractible iff $\mathsf{fib}_{f(x)}\left(v\right)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\left(x\right) \to Q\left(x\right)$ is a fiberwise equivalence
- Iff for all $x : A$, $f\left(x\right) : P\left(x\right) \to Q\left(x\right)$ is an equivalence.
- Iff for all $x : A$, $f\left(x\right)$ is contractible.
- Iff for all $x : A$ and for all $v \in Q\left(x\right)$, $\mathsf{fib}_{f(x)}\left(v\right)$ is contractible.
- Iff for all $w : \sum_{(x:A)} Q\left(x\right)$, $\mathsf{fib}_{\mathsf{total}(f)}\left(w\right)$ is contractible.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\,(x) \to Q\,(x)$ is a fiberwise equivalence
- Iff for all $x : A,\ f\,(x) : P\,(x) \to Q\,(x)$ is an equivalence.
- Iff for all $x : A,\ f\,(x)$ is contractible.
- Iff for all $x : A$ and for all $v \in Q\,(x)\,,\ \mathsf{fib}_{f(x)}\ (v)$ is contractible.
- Iff for all $w : \sum_{(x:A)} Q\,(x)\,, \mathsf{fib}_{\mathsf{total}(f)}\ (w)$ is contractible.
- Iff $\mathsf{total}\,(f)$ is contractible.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\,(x) \to Q\,(x)$ is a fiberwise equivalence
- Iff for all $x : A$, $f\,(x) : P\,(x) \to Q\,(x)$ is an equivalence.
- Iff for all $x : A$, $f\,(x)$ is contractible.
- Iff for all $x : A$ and for all $v \in Q\,(x)$, $\mathsf{fib}_{f(x)}\ (v)$ is contractible.
- Iff for all $w : \sum_{(x:A)} Q\,(x)$, $\mathsf{fib}_{\mathsf{total}(f)}\ (w)$ is contractible.
- Iff $\mathsf{total}\,(f)$ is contractible.
- Iff $\mathsf{total}\,(f)$ is an equivalence.

proof of the Theorem 4.7.7.

By Theorem 4.7.6 it follows for all $x : A$ and $v : Q(x)$ that

$$\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v)) \simeq \mathsf{fib}_{f(x)}\ (v)\,.$$

Equivalently, $\mathsf{fib}_{\mathsf{total}(f)}\ (\mathsf{pair}\,(x, v))$ is contractible iff $\mathsf{fib}_{f(x)}\ (v)$ is contractible.

We can trivially observe that:

- $f : \prod_{(x:A)} P\,(x) \to Q\,(x)$ is a fiberwise equivalence
- Iff for all $x : A$, $f\,(x) : P\,(x) \to Q\,(x)$ is an equivalence.
- Iff for all $x : A$, $f\,(x)$ is contractible.
- Iff for all $x : A$ and for all $v \in Q\,(x)$, $\mathsf{fib}_{f(x)}\ (v)$ is contractible.
- Iff for all $w : \sum_{(x:A)} Q\,(x)$, $\mathsf{fib}_{\mathsf{total}(f)}\ (w)$ is contractible.
- Iff $\mathsf{total}\,(f)$ is contractible.
- Iff $\mathsf{total}\,(f)$ is an equivalence.

$\square$

The Main Theorem

"I prefer Long Proofs to Short Proofs, the same way that I prefer long walks in the woods to short ones".

Vladimir Voevodsky (quoted by Avi Wigderson, Memorial Service for VV, Oct. 8, 2017, at IAS).

The Main Theorem

"I prefer Long Proofs to Short Proofs, the same way that I prefer long walks in the woods to short ones".

Vladimir Voevodsky (quoted by Avi Wigderson, Memorial Service for VV, Oct. 8, 2017, at IAS).

We will brake the proof into shorter proofs...

Weak Function Extensionality Principle

Definition (WFE)

The **weak function extensionality principle** asserts that there is a function

$$\prod_{x:A} \mathsf{IsContr}\,(P\,(x)) \to \mathsf{IsContr}\left(\prod_{x:A} P\,(x)\right)$$

for any family $P : A \to \mathcal{U}$ of types over any type A.

Weak Function Extensionality Principle

Definition (WFE)

The **weak function extensionality principle** asserts that there is a function

$$\prod_{x:A} \mathsf{IsContr}(P(x)) \rightarrow \mathsf{IsContr}\left(\prod_{x:A} P(x)\right)$$

for any family $P : A \rightarrow \mathcal{U}$ of types over any type A.

Lemma (Lemma 4.9.2.)

Assuming $\mathcal{U}$ is univalent, for any $A, B, X : \mathcal{U}$ and any $e : A \simeq B$, there is an equivalence

$$(X \rightarrow A) \simeq (X \rightarrow B)$$

of which the underlying map is given by post-composition with the underlying function of e.

proof of the Lemma 4.9.2.

Let $e : A \simeq B$. By induction we may assume that $e :\equiv (f_e, \alpha)$, where $f_e : A \to B$ and $\alpha : \mathsf{IsEquiv}(f_e)$.

proof of the Lemma 4.9.2.

Let $e : A \simeq B$. By induction we may assume that $e :\equiv (f_e, \alpha)$, where $f_e : A \to B$ and $\alpha : \mathsf{IsEquiv}(f_e)$.

Let us assume the map given by post-composition with the underlying function of e

$$\lambda\,(g : X \to A)\,.g \circ f_e : (X \to A) \to (X \to B)\,.$$

proof of the Lemma 4.9.2.

Let $e : A \simeq B$. By induction we may assume that $e :\equiv (f_e, \alpha)$, where $f_e : A \to B$ and α :$\mathsf{IsEquiv}(f_e)$.

Let us assume the map given by post-composition with the underlying function of e

$$\lambda\,(g : X \to A)\,.g \circ f_e : (X \to A) \to (X \to B)\,.$$

As $e : A \simeq B$, by UA we have that

$$\mathsf{idtoeqv}\colon (A = B) \to (A \simeq B)$$

is an equuivalence and thus we may assume that e is of the form $\mathsf{idtoeqv}(p)$, for some $p : A = B$; i.e.

$$e = \mathsf{idtoeqv}\,(p)\,.$$

proof of the Lemma 4.9.2. (Cont'd)

By path induction, assuming $p :\equiv \mathsf{refl}_A$, we have

$$e = \mathsf{idtoeqv}(\mathsf{refl}_A) \equiv e = \mathsf{transport}^{X \mapsto X}(\mathsf{refl}_A, -) \equiv e = \mathsf{id}_A \ .$$

proof of the Lemma 4.9.2. (Cont'd)

By path induction, assuming $p :\equiv \mathsf{refl}_A$, we have

$$e = \mathsf{idtoeqv}(\mathsf{refl}_A) \equiv e = \mathsf{transport}^{X \mapsto X}(\mathsf{refl}_A, -) \equiv e = \mathsf{id}_A \ .$$

Thus we have

$$\lambda(g : X \to A).g \circ f_e = \lambda(g : X \to A).g \circ \mathsf{id}_A \equiv \lambda(g : X \to A).g \circ f_e = \mathsf{id}_{(X \to A) \to (X \to A)}$$

which $\mathsf{id}_{(X \to A) \to (X \to A)}$ is an equivalence of $(X \to A) \simeq (X \to A)$.

$\square$

Corollary (Corollary 4.9.3.)

Let $P : A \to \mathcal{U}$ *be a family of contractible types,*

Corollary (Corollary 4.9.3.)

Let $P : A \to \mathcal{U}$ be a family of contractible types, i.e. $\prod_{(x:A)} \mathsf{IsContr}(P(x))$. Then the projection $\mathsf{pr}_1 \colon \left(\sum_{(x:A)} P(x)\right) \to A$ is an equivalence.

Corollary (Corollary 4.9.3.)

Let $P : A \to \mathcal{U}$ be a family of contractible types, i.e. $\prod_{(x:A)} \mathsf{IsContr}\,(P\,(x))$. Then the projection pr_1: $\left(\sum_{(x:A)} P\,(x)\right) \to A$ is an equivalence.

Assuming $\mathcal{U}$ is univalent, it follows immediately that post-composition with pr_1 gives an equivalence

$$\alpha : \left(A \to \sum_{x:A} P\,(x)\right) \simeq (A \simeq A)\,.$$

proof of the Corollary 4.9.3.

By Lemma 4.8.1 , for $\mathsf{pr}_1 \colon \left(\sum_{(x:A)} P(x)\right) \to A$ and $x : A$ we have an equivalence
$$\mathsf{fib}_{\mathsf{pr}_1}(x) \simeq P(x).$$

proof of the Corollary 4.9.3.

By Lemma 4.8.1 , for $\mathsf{pr}_1\colon \left(\sum_{(x:A)} P(x)\right) \to A$ and $x : A$ we have an equivalence

$$\mathsf{fib}_{\mathsf{pr}_1}(x) \simeq P(x)\,.$$

As for any $x : A$ we have that $P(x)$ is contractible, we get that pr_1 is contractible, or equivalently pr_1 is an equivalence of

$$\left(\sum_{(x:A)} P(x)\right) \simeq A$$

proof of the Corollary 4.9.3.

By Lemma 4.8.1 , for pr_1: $\left(\sum_{(x:A)} P(x)\right) \to A$ and $x : A$ we have an equivalence

$$\mathsf{fib}_{\mathsf{pr}_1}(x) \simeq P(x).$$

As for any $x : A$ we have that $P(x)$ is contractible, we get that pr_1 is

contractible, or equivalently pr_1 is an equivalence of

$$\left(\sum_{(x:A)} P(x)\right) \simeq A$$

By Lemma 4.9.2. for $X :\equiv A$ we have

$$\left(A \to \sum_{x:A} P(x)\right) \simeq (A \simeq A)$$

as wanted.

$\square$

UA implies WFE

Theorem (Theorem 4.9.4.)

In a univalent universe $\mathcal{U}$, suppose that $P : A \to \mathcal{U}$ is a family of contractible types and let

$$\alpha : \left(A \to \sum_{x:A} P(x) \right) \simeq (A \simeq A).$$

Then $\prod_{(x:A)} P(x)$ is a retract of $\mathsf{fib}_\alpha(\mathsf{id}_A)$.
As a consequence, $\prod_{(x:A)} P(x)$ is contractible.

UA implies WFE

Theorem (Theorem 4.9.4.)

In a univalent universe $\mathcal{U}$, suppose that $P : A \to \mathcal{U}$ is a family of contractible types and let

$$\alpha : \left(A \to \sum_{x:A} P(x) \right) \simeq (A \simeq A).$$

Then $\prod_{(x:A)} P(x)$ is a retract of $\mathsf{fib}_\alpha(\mathsf{id}_A)$.
As a consequence, $\prod_{(x:A)} P(x)$ is contractible.

In other words, the univalence axiom implies the weak function extensionality principle.

proof of the Theorem 4.9.4.

We define the following functions:

proof of the Theorem 4.9.4.

We define the following functions:

Section

$$\varphi : \prod_{(x:A)} P(x) \to \mathsf{fib}_{\alpha}(\mathsf{id}_A)$$

$$\varphi(f) :\equiv \mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)$$

proof of the Theorem 4.9.4.

We define the following functions:

Section

$$\varphi : \prod_{(x:A)} P(x) \to \mathsf{fib}_{\alpha}(\mathsf{id}_A)$$

$$\varphi(f) :\equiv \mathsf{pair}\big(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\big)$$

Observe that it φ well defined:

proof of the Theorem 4.9.4.

We define the following functions:

Section

$$\varphi : \prod_{(x:A)} P(x) \to \mathsf{fib}_{\alpha}(\mathsf{id}_A)$$

$$\varphi(f) :\equiv \mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)$$

Observe that it φ well defined:

- $\lambda(x:A).\,\mathsf{pair}(x, f(x)) : A \to \sum_{x:A} P(x)$

proof of the Theorem 4.9.4.

We define the following functions:

Section

$$\varphi : \prod_{(x:A)} P(x) \to \mathsf{fib}_{\alpha}(\mathsf{id}_A)$$

$$\varphi(f) :\equiv \mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)$$

Observe that it φ well defined:

- $\lambda(x:A).\,\mathsf{pair}(x, f(x)) : A \to \sum_{x:A} P(x)$
- $\mathsf{fib}_{\alpha}(\mathsf{id}_A) \equiv \sum_{(z:A\to\sum_{x:A} P(x))} \alpha(z) = \mathsf{id}_A$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_{\alpha}(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g,p)) :\equiv \lambda(x:A).\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))$$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_{\alpha}(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g, p)) :\equiv \lambda(x : A).\,\mathsf{happly}(p, x)_* (\mathsf{pr}_2(g(x)))$$

Observe that ψ is well defined:

- $g : A \to \sum_{(x:A)} P(x)$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_{\alpha}(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g,p)) :\equiv \lambda(x:A).\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))$$

Observe that ψ is well defined:

- $g : A \to \sum_{(x:A)} P(x)$
- $p : \alpha(g) = \mathsf{id}_A$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_{\alpha}(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g,p)) :\equiv \lambda(x:A).\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))$$

Observe that ψ is well defined:

- $g : A \to \sum_{(x:A)} P(x)$
- $p : \alpha(g) = \mathsf{id}_A$
- $\mathsf{happly}(p,-) : \prod_{(x:A)} \alpha(g)(x) = x$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_{\alpha}(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g,p)) :\equiv \lambda(x:A).\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))$$

Observe that ψ is well defined:

- $g : A \to \sum_{(x:A)} P(x)$
- $p : \alpha(g) = \mathsf{id}_A$
- $\mathsf{happly}(p,-) : \prod_{(x:A)} \alpha(g)(x) = x$
- $\mathsf{happly}(p,x)_* : P(\alpha(g)(x)) \to P(x)$

proof of the Theorem 4.9.4. (Cont'd)

Retraction

$$\psi : \mathsf{fib}_\alpha(\mathsf{id}_A) \to \prod_{x:A} P(x)$$

$$\psi(\mathsf{pair}(g,p)) :\equiv \lambda(x:A).\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x)))$$

Observe that ψ is well defined:

- $g : A \to \sum_{(x:A)} P(x)$
- $p : \alpha(g) = \mathsf{id}_A$
- $\mathsf{happly}(p,-) : \prod_{(x:A)} \alpha(g)(x) = x$
- $\mathsf{happly}(p,x)_* : P(\alpha(g)(x)) \to P(x)$
- $\lambda x.\,\mathsf{happly}(p,x)_*(\mathsf{pr}_2(g(x))) : \prod_{(x:A)} P(x)$

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\psi \circ \varphi(f) \equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right)$$

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}\psi \circ \varphi(f) &\equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right)\\ &\equiv \lambda(x:A).\,\mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_*(f(x))\end{aligned}$$

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}
\psi \circ \varphi(f) &\equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right) \\
&\equiv \lambda(x:A).\,\mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_*(f(x)) \\
&\equiv \lambda(x:A).f(x)
\end{aligned}$$

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}
\psi \circ \varphi (f) &\equiv \psi \left(\mathsf{pair}\left(\lambda (x:A) \,.\, \mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right) \\
&\equiv \lambda (x:A) \,.\, \mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_* (f(x)) \\
&\equiv \lambda (x:A) \,.\, f(x) \\
&\equiv f
\end{aligned}$$

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}
\psi \circ \varphi(f) &\equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right)\\
&\equiv \lambda(x:A).\,\mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_*(f(x))\\
&\equiv \lambda(x:A).f(x)\\
&\equiv f
\end{aligned}$$

Thus $\prod_{(x:A)} P(x)$ is a retract of $\mathsf{fib}_\alpha(\mathsf{id}_A)$.

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}
\psi \circ \varphi(f) &\equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right) \\
&\equiv \lambda(x:A).\,\mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_*(f(x)) \\
&\equiv \lambda(x:A).f(x) \\
&\equiv f
\end{aligned}$$

Thus $\prod_{(x:A)} P(x)$ is a retract of $\mathsf{fib}_\alpha(\mathsf{id}_A)$.

But from Corollary 4.9.3. $\mathsf{fib}_\alpha(\mathsf{id}_A)$ is contractible.

proof of the Theorem 4.9.4. (Cont'd)

Let $f : \prod_{(x:A)} P(x)$.
We have that

$$\begin{aligned}
\psi \circ \varphi(f) &\equiv \psi\left(\mathsf{pair}\left(\lambda(x:A).\,\mathsf{pair}(x, f(x)), \mathsf{refl}_{\mathsf{id}_A}\right)\right) \\
&\equiv \lambda(x:A).\,\mathsf{happly}\left(\mathsf{refl}_{\mathsf{id}_A}, x\right)_*(f(x)) \\
&\equiv \lambda(x:A).f(x) \\
&\equiv f
\end{aligned}$$

Thus $\prod_{(x:A)} P(x)$ is a retract of $\mathsf{fib}_\alpha(\mathsf{id}_A)$.

But from Corollary 4.9.3. $\mathsf{fib}_\alpha(\mathsf{id}_A)$ is contractible.

Therefore by Lemma 3.11.7. we conclude that $\mathsf{fib}_\alpha(\mathsf{id}_A)$ is contractible.
$\square$

WFE implies FunExt

Theorem (Theorem 4.9.5.)

Weak function extensionality implies the function extensionality Axiom.

WFE implies FunExt

Theorem (Theorem 4.9.5.)

Weak function extensionality implies the function extensionality Axiom.

Therefore

UA implies FunExt

proof of Theorem 4.9.5.

We want to show that the type

$$\prod_{A:\mathcal{U}} \prod_{P:A\to\mathcal{U}} \prod_{f,g:\prod_{(x:A)} P(x)} \mathsf{IsEquiv}(\mathsf{happly}(f,g))$$

is inhabited.

proof of Theorem 4.9.5.

We want to show that the type

$$\prod_{A:\mathcal{U}} \prod_{P:A \to \mathcal{U}} \prod_{f,g:\prod_{(x:A)} P(x)} \mathsf{IsEquiv}(\mathsf{happly}(f,g))$$

is inhabited.

It suffices to show that

$$\lambda \left(g : \prod_{x:A} P(x) \right) . \, \mathsf{happly}(f,g) : \prod_{g:\prod_{(x:A)} P(x)} ((f = g) \to (f \sim g))$$

is a fiberwise equivalence.

proof of Theorem 4.9.5. (Cont'd)

Since a fiberwise map induces an equivalence on total spaces iff it is fiberwise an equivalence by Theorem 4.7.7 , where we assume

- $A \leftarrow \prod_{(s:A)} P(x)$
- $P(x) \leftarrow f = g$
- $Q(x) \leftarrow f \sim g$
- $f \leftarrow \lambda \left(g : \prod_{x:A} P(x)\right) . \operatorname{happly}(f, g)$

it suffices to show that the function

$$\operatorname{total}\left(\lambda\left(g : \prod_{x:A} P(x)\right) . \operatorname{happly}(f, g)\right) : \sum_{g:\prod_{(x:A)} P(x)} (f = g) \to \sum_{g:\prod_{(x:A)} P(x)} (f \sim g))$$

is an equivalence.

proof of Theorem 4.9.5. (Cont'd)

By Lemma 3.11.8. we know that $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f = g)$ is contractible.

proof of Theorem 4.9.5. (Cont'd)

By Lemma 3.11.8. we know that $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f = g)$ is contractible.

It suffices to show that the type$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g)$ is also contractible.

proof of Theorem 4.9.5. (Cont'd)

By Lemma 3.11.8. we know that $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f = g)$ is contractible.

It suffices to show that the type$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g)$ is also contractible.

?!

proof of Theorem 4.9.5. (Cont'd)

By Lemma 3.11.8. we know that $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f = g)$ is contractible.

It suffices to show that the type $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g)$ is also contractible.

?!

> *"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"*
>
> *Vladimir Voevodsky [3]*

Our Lemma

Lemma

Suppose function $f : A \to B$. If the types A, B are contractible, then f is an equivalence.

Our Lemma

Lemma

Suppose function $f : A \to B$. If the types A, B are contractible, then f is an equivalence.

proof of Lemma

Let $a : A$ and $b : B$ the corresponding centers of contraction;

Our Lemma

Lemma

Suppose function $f : A \to B$. If the types A, B are contractible, then f is an equivalence.

proof of Lemma

Let $a : A$ and $b : B$ the corresponding centers of contraction; i.e.

- $\alpha : \mathsf{IsContr}\ (A)$ and $a :\equiv \mathsf{pr}_1\ (\alpha)$
- $\beta : \mathsf{IsContr}\ (B)$ and $b :\equiv \mathsf{pr}_1\ (\beta)$

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

Thus $(a, p_y) : \mathsf{fib}_f(y)$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

Thus $(a, p_y) : \mathsf{fib}_f(y)$. We want to show that (a, p_y) is center of retraction of $\mathsf{fib}_f(y)$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

Thus $(a, p_y) : \mathsf{fib}_f(y)$. We want to show that (a, p_y) is center of retraction of $\mathsf{fib}_f(y)$.

Let $w : \mathsf{fib}_f(y)$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

Thus $(a, p_y) : \mathsf{fib}_f(y)$. We want to show that (a, p_y) is center of retraction of $\mathsf{fib}_f(y)$.

Let $w : \mathsf{fib}_f(y)$. By induction for Σ-types we may assume that $w :\equiv (a', p')$.
We want to show that $(a, p_y) = (a', p')$.

proof of Lemma (Cont'd)

As B is contractible there are

- $p : b = f(a)$
- $q_y :\equiv \mathsf{pr}_2(\beta)(y) : b = y$, for any $y : B$.

Let us fix $y : B$. We define

$$p_y :\equiv p^{-1} \cdot q_y : f(a) = y.$$

Thus $(a, p_y) : \mathsf{fib}_f(y)$. We want to show that (a, p_y) is center of retraction of $\mathsf{fib}_f(y)$.

Let $w : \mathsf{fib}_f(y)$. By induction for Σ-types we may assume that $w :\equiv (a', p')$.
We want to show that $(a, p_y) = (a', p')$.
By Theorem 2.7.2. it suffices to show that

$$\sum_{k:a=a'} \mathsf{transport}^{\mathsf{fib}_f(y)}(k, p_y) = p'.$$

proof of Lemma (Cont'd)

We have $\mathsf{pr}_2(\alpha)(a') : a = a'$.

proof of Lemma (Cont'd)

We have $\mathsf{pr}_2(\alpha)(a') : a = a'$.

By path induction we may assume $a' :\equiv a$,

proof of Lemma (Cont'd)

We have $\mathsf{pr}_2\,(\alpha)\,(a') : a = a'$.

By path induction we may assume $a' :\equiv a$, thus it suffices to show that

$$\mathsf{transport}^{\mathsf{fib}_f(y)}\,(\mathsf{refl}_a, p_y) = p' \equiv p_y = p'.$$

proof of Lemma (Cont'd)

We have $\mathsf{pr}_2(\alpha)\left(a'\right) : a = a'$.

By path induction we may assume $a' :\equiv a$, thus it suffices to show that

$$\mathsf{transport}^{\mathsf{fib}_f(y)}\left(\mathsf{refl}_a, p_y\right) = p' \equiv p_y = p'.$$

By path induction we may assume also that $y :\equiv f(a)$.

proof of Lemma (Cont'd)

We have $\mathsf{pr}_2(\alpha)\left(a'\right) : a = a'$.

By path induction we may assume $a' :\equiv a$, thus it suffices to show that

$$\mathsf{transport}^{\mathsf{fib}_f(y)}\left(\mathsf{refl}_a, p_y\right) = p' \equiv p_y = p'.$$

By path induction we may assume also that $y :\equiv f(a)$. Thus it

suffices to show that

$$p_{f(a)} = p' \equiv \mathsf{refl}_{f(a)} = \mathsf{refl}_{f(a)}$$

which is inhabited by $\mathsf{refl}_{\mathsf{refl}_{f(a)}}$.

$\square$

proof of Theorem 4.9.5. (Cont'd)

Now by Theorem 2.15.7 / Θεώρημα 58 (aka AC) we get that

$$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g) \text{ is a retract of } \prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$$

(Without using FunExt).

proof of Theorem 4.9.5. (Cont'd)

Now by Theorem 2.15.7 / Θεώρημα 58 (aka AC) we get that

$$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g) \text{ is a retract of } \prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$$

(Without using FunExt).

By Lemma 3.11.8. we can observe that $\prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$ is a product of contractible types.

proof of Theorem 4.9.5. (Cont'd)

Now by Theorem 2.15.7 / Θεώρημα 58 (aka AC) we get that

$$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g) \text{ is a retract of } \prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$$

(Without using FunExt).

By Lemma 3.11.8. we can observe that $\prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$ is a product of contractible types.

Thus by WFE we get that $\prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$.

proof of Theorem 4.9.5. (Cont'd)

Now by Theorem 2.15.7 / Θεώρημα 58 (aka AC) we get that

$$\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g) \text{ is a retract of } \prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$$

(Without using FunExt).

By Lemma 3.11.8. we can observe that $\prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$ is a product of contractible types.

Thus by WFE we get that $\prod_{(x:A)} \sum_{(u:P(x))} (f(x) = u)$.

Therefore, by Lemma 3.11.7. we have that $\sum_{\left(g:\prod_{(x:A)} P(x)\right)} (f \sim g)$ is also contractible, as wanted.

$\square$

This proof was discovered by the one and only Vladimir Voevodsky!

This proof was discovered by the one and only Vladimir Voevodsky!

He proved it using Coq!

This proof was discovered by the one and only Vladimir Voevodsky!

He proved it using Coq!

```
Proof.
  ∃ (fun p ⇒ snd (projT1 p)).
  intros x.
  eexists (existT _ (existT (fun (xy : A × A) ⇒ fst xy ⇝ snd xy) (x,x) (idpath x)) _).
  intros [[[u v] p] q].
  simpl in × ⊢ ×.
  induction q as [a].
  induction p as [b].
  apply idpath.
Defined.
```

And finally, we are ready to prove that extensionality of maps holds, i.e., if two maps are pointwise homotopic then they are homotopic. First we outline the proof.

Suppose maps $f\ g : A \to B$ are extensionally equal via a pointwise homotopy p. We seek a path $f \leadsto g$. Because $eta\ f \leadsto f$ and $eta\ g \leadsto g$ it suffices to find a path $eta\ f \leadsto eta\ g$.

Consider the maps $d\ e : S \to path\_space\ T$ where $d\ x = existT\ \_\ (f\ x, f\ x)\ (idpath\ x)$ and $e\ x = existT\ \_\ (f\ x, g\ x)\ (p\ x)$. If we compose d and e with trg we get $eta\ f$ and $eta\ g$, respectively. So, if we had a path from d to e, we would get one from $eta\ f$ to $eta\ g$. But we can get a path from d to e because $src \circ d = eta\ f = src \circ e$ and composition with src is an equivalence.

```
Theorem extensionality {A B : Set} (f g : A → B) : (∀ x, f x ⇝ g x) → (f ⇝ g).
Proof.
  intro p.
  pose (d := fun x : A ⇒ existT (fun xy ⇒ fst xy ⇝ snd xy) (f x, f x) (idpath (f x))).
  pose (e := fun x : A ⇒ existT (fun xy ⇒ fst xy ⇝ snd xy) (f x, g x) (p x)).
  pose (src_compose := weq_exponential (src B) A).
  pose (trg_compose := weq_exponential (trg B) A).
  apply weq_injective with (w := etaweq A B).
  simpl.
  path_via (projT1 trg_compose e).
  path_via (projT1 trg_compose d).
  apply map.
  apply weq_injective with (w := src_compose).
  apply idpath.
Defined.
```

And that is all, thank you.

A Coq proof that Univalence Axioms implies Functional Extensionality

Andrej Bauer, Peter LeFanu Lumsdaine

Bibliography

- The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, https://homotopytypetheory.org/book/. Institute for Advanced Study, 2013.
- Martin Hofmann and Thomas Streicher. *The groupoid interpretation of type theory.* In Giovanni Sambin and Jan M. Smith, editors, *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of Oxford Logic Guides, pages 83–111. Oxford University Press, New York, 1998.
- Vladimir Voevodsky. *UNIVALENT FOUNDATIONS* , Institute for Advanced Study Princeton, NJ March 26, 2014