Axioms for Univalence

Ian Orton
(joint work with Andrew Pitts)

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

TYPES, Budapest, 2017
The identity type
The identity type

We have a type:

\[
\begin{align*}
\Gamma & \vdash A \\
\Gamma & \vdash a_0 : A \\
\Gamma & \vdash a_1 : A \\
\hline
\Gamma & \vdash a_0 = a_1
\end{align*}
\]
The identity type

We have a type:

\[ \Gamma \vdash A \quad \Gamma \vdash a_0 : A \quad \Gamma \vdash a_1 : A \]

\[ \Gamma \vdash a_0 = a_1 \]

With the introduction rule:

\[ \Gamma \vdash A \quad \Gamma \vdash a : A \]

\[ \Gamma \vdash \text{refl} : a = a \]
Equational reasoning

Given a proofs $a_0 = a_1$ and $a_1 = a_2$ we can deduce

$$a_0 = a_2$$
Equational reasoning

Given a proofs $a_0 = a_1$ and $a_1 = a_2$ we can deduce

$$a_0 = a_2$$

This means that we can use equational reasoning, e.g.

$$(n + 0) \times 2 = n \times 2$$
$$= 2 \times n$$
$$= (5 - 3) \times n$$
Contractibility

Given a type $A$ consider the following property:
Contractibility

Given a type \( A \) consider the following property:

\[
isContr(A) \triangleq \sum_{a_0:A} \prod_{a:A} (a_0 = a)
\]
Contractibility

Given a type $A$ consider the following property:

$$isContr(A) \triangleq \sum_{a_0:A} \prod_{a:A} (a_0 = a)$$

An element of $isContr(A)$ is a pair $(a_0, ctr)$ where

$$a_0 : A$$
Contractibility

Given a type $A$ consider the following property:

$$isContr(A) \triangleq \sum_{a_0:A} \prod_{a:A} (a_0 = a)$$

An element of $isContr(A)$ is a pair $(a_0, ctr)$ where

$$a_0 : A$$

and, for any other $a : A$, we have

$$ctr(a) : a_0 = a$$
Contractibility

Given a type $A$ consider the following property:

$$\text{isContr}(A) \triangleq \sum_{a_0:A} \prod_{a:A} (a_0 = a)$$

An element of $\text{isContr}(A)$ is a pair $(a_0, \text{ctr})$ where

$$a_0 : A$$

and, for any other $a : A$, we have

$$\text{ctr}(a) : a_0 = a$$

Therefore there exists a unique term of type $A$. 
Singletons are contractible

One example of a contractible type is the type of singletons. Given $a : A$ we define
Singletons are contractible

One example of a contractible type is the type of singletons. Given \( a : A \) we define

\[
sing(a) \triangleq \sum_{a' : A} (a = a')
\]
Singletons are contractible

One example of a contractible type is the type of singletons. Given $a : A$ we define

$$\text{sing}(a) \triangleq \sum_{a' : A} (a = a')$$

We can show that $\text{sing}(a)$ is contractible for all $A : \mathcal{U}$ and $a : A$. 
Equivalences

What does it mean for two types $A$ and $B$ to be equivalent?
Equivalences

What does it mean for two types \( A \) and \( B \) to be equivalent?

Two sets \( X \) and \( Y \) are isomorphic if there exists a function \( f : X \rightarrow Y \) such that

\[
\forall y \in Y. \exists！x \in X. f(x) = y
\]
Equivalences

What does it mean for two types $A$ and $B$ to be equivalent?

Two sets $X$ and $Y$ are isomorphic if there exists a function $f : X \rightarrow Y$ such that

$$\forall y \in Y. \exists! x \in X. f(x) = y$$

We want to express this in type theory.
Equivalences

Given a map \( f : A \to B \) define:

\[
\text{fib}_f(b) \triangleq \sum_{a:A} (f \ a = b)
\]
Equivalences

Given a map \( f : A \to B \) define:

\[
\text{fib}_f(b) \triangleq \sum_{a:A} (f \ a = b)
\]

We say that a map \( f : A \to B \) is an equivalence if \( \text{fib}_f(b) \) is contractible for all \( b : B \).
Equivalences

Given a map \( f : A \to B \) define:

\[
\text{fib}_f(b) \triangleq \sum_{a:A} (f a = b)
\]

We say that a map \( f : A \to B \) is an equivalence if \( \text{fib}_f(b) \) is contractible for all \( b : B \).

We write \( A \simeq B \) for the type of equivalences from \( A \) to \( B \).
Homotopy Type Theory

Think of a type $A$ as a space with terms $a : A$ as points and equality proofs $p : a_0 = a_1$ as paths from $a_0$ to $a_1$. 
Homotopy Type Theory

Think of a type $A$ as a space with terms $a : A$ as points and equality proofs $p : a_0 = a_1$ as paths from $a_0$ to $a_1$.

What about equality between types $A$ and $B$? What does it mean to say that $A = B$?
Think of a type $A$ as a space with terms $a : A$ as points and equality proofs $p : a_0 = a_1$ as paths from $a_0$ to $a_1$.

What about equality between types $A$ and $B$? What does it mean to say that $A = B$?

It means that the two spaces are homotopic. Captured by the univalence axiom.
Univalence

The univalence axiom states that

$$(A = B) \simeq (A \simeq B)$$

for all $A, B : \mathcal{U}$. 
The univalence axiom states that

\[(A = B) \simeq (A \simeq B)\]

for all \(A, B : \mathcal{U}\).\(^1\)

---

1. Where one half of the equivalence is the canonical map \((A = B) \rightarrow (A \simeq B)\)
Coerce

For any two types $A$ and $B$ we can define a function

$$coerce : (A = B) \rightarrow A \rightarrow B$$
Univalence revisited

The univalence axiom is satisfied iff we have a map

$$\text{ua} : (A \simeq B) \to A = B$$

such that

$$\text{coerce}(\text{ua}(f, e)) = f$$

for all $A, B : \mathcal{U}$. 
Univalence revisited

The univalence axiom is satisfied iff we have a map

$$ua : (A \simeq B) \to A = B$$

such that

$$\text{coerce}(ua(f, e)) = f$$

for all $A, B : \mathcal{U}$. \(^2\)

---

2. Due to Dan Licata: “weak univalence with "beta" implies full univalence”, HoTT mailing list.
The Axioms (1)

We assume function extensionality and:

$$\text{unit} : A = \sum_{a : A} 1$$

such that

$$\text{coerce unit } a = (a, *)$$

for all $$A : \mathcal{U}$$.
The Axioms (2)

We assume function extensionality and:

\[ \text{flip : } \sum_{a:A} \sum_{b:B} C \ a \ b = \sum_{b:B} \sum_{a:A} C \ a \ b \]

such that

\[ \text{coerce flip} \ (a, b, c) = (b, a, c) \]

for all \( A, B : \mathcal{U} \) and \( C : A \to B \to \mathcal{U} \).
We assume function extensionality and:

\[
\text{contract} : \text{isContr } A \rightarrow A = 1
\]

for all \( A : U \).
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(ua(f) : A = B\) like so:
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(u a(f) : A = B\) like so:

\[ A \]
Assume that we are given \((f, e) : A \simeq B\) and define \(u_a(f) : A = B\) like so:

\[
A = \sum_{a : A} 1
\]
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(ua(f) : A = B\) like so:

\[
A = \sum_{a : A} 1 = \sum_{a : A} \sum_{b : B} f(a) = b
\]

\(\text{sing}(f \ a)\)
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(u a(f) : A = B\) like so:

\[
A = \sum_{a:A} 1
\]

\[
= \sum_{a:A} \sum_{b:B} f \ a = b
\]

\[
= \sum_{b:B} \sum_{a:A} f \ a = b
\]
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(u a(f) : A = B\) like so:

\[
A = \sum_{a : A} 1
= \sum_{a : A} \sum_{b : B} f\ a = b
= \sum_{b : B} \sum_{a : A} f\ a = b
\]

\(f_{ib_f}(b)\)
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(ua(f) : A = B\) like so:

\[
A = \sum_{a : A} 1 = \sum_{a : A} \sum_{b : B} f \ a = b = \sum_{b : B} \sum_{a : A} f \ a = b = \sum_{b : B} 1
\]
Equality from equivalence

Assume that we are given \((f, e) : A \simeq B\) and define \(u_a(f) : A = B\) like so:

\[
A = \sum_{a:A} 1 \\
= \sum_{a:A} \sum_{b:B} f a = b \\
= \sum_{b:B} \sum_{a:A} f a = b \\
= \sum_{b:B} 1 \\
= B
\]
Coercing with univalence

Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(ua(f, e) : A = B\) as indicated previously.
Coercing with univalence

Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(ua(f, e) : A = B\) as indicated previously.

\[ a : A \]
Coercing with univalence

Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(u a(f, e) : A = B\) as indicated previously.

\[ a : A \mapsto (a, *) : \sum_{a : A} 1 \]
Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(ua(f, e) : A = B\) as indicated previously.

\[
a : A \mapsto (a, \ast)
\]

\[
\mapsto (a, f \ a, \text{refl})
\]

\[
: \sum_{a:A} \sum_{b:B} f \ a = b
\]
Coercing with univalence

Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(u_a(f, e) : A = B\) as indicated previously.

\[
\begin{align*}
a : A & \mapsto (a, *) \\
& \mapsto (a, f a, \text{refl}) \\
& \mapsto (f a, a, \text{refl})
\end{align*}
\]

\[
\begin{align*}
: & \sum_{a:A} 1 \\
& \sum_{a:A} \sum_{b:B} f a = b \\
& \sum_{b:B} \sum_{a:A} f a = b
\end{align*}
\]
Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(ua(f, e) : A = B\) as indicated previously.

\[
\begin{align*}
\forall a : A & \mapsto (a, \ast) : \sum_{a : A} 1 \\
\mapsto (a, f a, \text{refl}) & : \sum_{a : A} \sum_{b : B} f a = b \\
\mapsto (f a, a, \text{refl}) & : \sum_{b : B} \sum_{a : A} f a = b \\
\mapsto (f a, \ast) & : \sum_{b : B} 1
\end{align*}
\]
Coercing with univalence

Assume that we are given \((f, e) : A \simeq B\) and we have constructed \(ua(f, e) : A = B\) as indicated previously.

\[ a : A \mapsto (a, \ast) : \sum_{a : A} 1 \]
\[ \mapsto (a, f a, \text{refl}) : \sum_{a : A} \sum_{b : B} f a = b \]
\[ \mapsto (f a, a, \text{refl}) : \sum_{b : B} \sum_{a : A} f a = b \]
\[ \mapsto (f a, \ast) : \sum_{b : B} 1 \]
\[ \mapsto f a : B \]
Why is this better?

Given a type $A : \mathcal{U}$, write $[A]$ for the interpretation of $A$ in some model of HoTT. What happens in the model if $A \simeq B$?
Why is this better?

Given a type $A : \mathcal{U}$, write $[A]$ for the interpretation of $A$ in some model of HoTT. What happens in the model if $A \simeq B$?

We would expect to get morphisms:

$$f : [A] \to [B] \quad \text{and} \quad g : [B] \to [A]$$

which are not inverses up to equality, but rather up to some notion of homotopy.
Why is this better?

But what about the types $A$ and $A \times 1$?
Why is this better?

But what about the types $A$ and $A \times 1$? We would expect these types to yield two isomorphic objects in the model. That is, with

$$f : [A] \to [A \times 1] \quad \text{and} \quad g : [1 \times A] \to [A]$$

such that $g \circ f = id = f \circ g$. 
Why is this better?

But what about the types $A$ and $A \times 1$? We would expect these types to yield two isomorphic objects in the model. That is, with

$$f : \llbracket A \rrbracket \to \llbracket A \times 1 \rrbracket$$

and

$$g : \llbracket 1 \times A \rrbracket \to \llbracket A \rrbracket$$

such that $g \circ f = id = f \circ g$.

And similarly, we would expect:

$$\left[ \sum_{a:A} \sum_{b:B} C \ a \ b \right] \cong \left[ \sum_{b:B} \sum_{a:A} C \ a \ b \right]$$
Why is this better?

This means that we can satisfy *unit* and *flip* by proving that this stronger notion of isomorphism gives rise to a propositional equality between types.
Why is this better?

This means that we can satisfy \textit{unit} and \textit{flip} by proving that this stronger notion of isomorphism gives rise to a propositional equality between types.

Easy to do in the cubical sets model.
Why is this better?

So once we have this, we need function extensionality (often easy to check), leaving us with just the *contract* axiom.
Why is this better?

So once we have this, we need function extensionality (often easy to check), leaving us with just the *contract* axiom.

Also easy to do in the cubical sets model.
Thanks for listening!

Summary:

- Univalence can be reduced to a set of axioms that are potentially easier to check in models

"Axioms for Univalence"
Ian Orton and Andrew Pitts
Abstract and Agda: [http://www.cl.cam.ac.uk/~rio22/](http://www.cl.cam.ac.uk/~rio22/)
Ian.Orton@cl.cam.ac.uk Andrew.Pitts@cl.cam.ac.uk