Univalent fibrations in type theory and topology

Dan Christensen
University of Western Ontario

Wayne State University, April 11, 2016

Outline:

- Background on type theory
- Equivalence and univalence
- A characterization of univalent fibrations

A puzzle for the topologists

Let $p : E \to B$ be a Kan fibration with B a Kan complex. For each b_0, b_1 in B there is a map

$$\omega : \mathrm{Paths}(b_0, b_1) \to \mathrm{Equivalences}(E_{b_0}, E_{b_1}).$$

If ω is a weak equivalence for all b_0, b_1, we say that p is univalent.

Puzzle: Which p are univalent?

We'll come back to this at the end of the talk, after a long detour.

History of Type Theory

Initial ideas due to Bertrand Russell in early 1900's, to create a foundation for mathematics that avoids Russell's paradox.

Studied by many logicians and later computer scientists, particularly Church, whose λ-calculus (1930's and 40's) is a type theory.

In 1972, Per Martin-Löf extended type theory to include dependent types. It is this form of type theory that we will focus on. One of the key features is that it unifies set theory and logic!

In 2006, Awodey, Warren, and Voevodsky discovered that type theory has homotopical models, extending 1998 work of Hofmann and Streicher.

2012–2013: A special year at the IAS, which led to The HoTT book.

May 12-16, 2016: Come to our workshop at the Fields Institute!

Background on Type Theory

Type theory is a logical system in which the basic objects are called types.

Initially, types were thought of as sets, but we will see later that it is fruitful to think of them as being like spaces.

As in first order logic, type theory is a syntactic theory in which certain expressions are well-formed, and there are syntactic rules that tell you how to produce new expressions (i.e., theorems) from existing expressions.

First order logic can be used to study many theories: the theory of groups, Peano arithmetic, set theory (e.g., ZFC), etc.

In contrast, type theory is intrinsically a theory about sets/spaces. It is not a general framework for studying axiomatic systems, but instead unifies set theory and logic so that they live at the same level.

Background on Type Theory II

People study type theory for many reasons. I'll highlight two:

- Its intrinsic homotopical content. (This talk.)
- Its suitability for computer formalization. (Another talk!)

We'll dive right in without being formal about it.

We write $x : X$ to indicate that x is an element of type X, which is analogous to the set-theoretic statement $x \in X$.

Each element has a unique type, so we can't directly talk about intersections, unions, etc. Instead, type theory comes with type constructors that correspond to common constructions in mathematics.

We assume given a universe type `Type`, and therefore can write $X : \texttt{Type}$ to indicate that X is a type.

Type Constructors: Function types

For any two types A and B, there is a function type denoted $A \to B$, which should be thought of as an internal hom B^A.

If $f(a)$ is an expression of type B whenever a is of type A, then $\lambda a.f(a)$ denotes the function $A \to B$ sending a to $f(a)$.

Conversely, if $f : A \to B$ and $a : A$, then $f(a) : B$.

Examples:

- The identity function $\mathtt{id}_A$ is defined to be $\lambda a.a$.
- The constant function sending everything in A to $b : B$ is $\lambda a.b$.
- Given functions $f : A \to B$ and $g : B \to C$, their composite $gf : A \to C$ is $\lambda a.g(f(a))$.
- And $\lambda f.\lambda g.\lambda a.g(f(a))$ has type

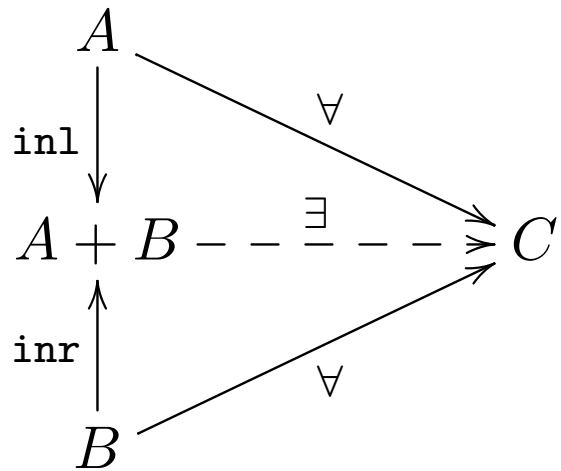
$$(A \to B) \to ((B \to C) \to (A \to C)).$$

Type Constructors: Coproduct

Most constructions in type theory are defined inductively.

For example, given types A and B, there is another type $A + B$ which is generated by elements of the form `inl` a and `inr` b.

"Generated" means that it satisfies a weak universal property:



Type Constructors: $\varnothing$, $\mathbf{1}$, $\times$, $\mathbb{N}$

Here are other types defined by such induction principles:

- The empty type $\varnothing$ is a weakly initial object ("free on no generators"): for any C, there is a map $\varnothing \to C$.
- The one point type $\mathbf{1}$ is "free on one generator $*$": given $c : C$, there is a map $f : \mathbf{1} \to C$ with $f(*) = c$.
- The product $A \times B$ of two types is generated by all pairs (a, b): given $g : A \to (B \to C)$, we get $f : A \times B \to C$ with $f(a, b) = g(a)(b)$.
- The type of natural numbers $\mathbb{N}$ is generated by $0 : \mathbb{N}$ and $\mathtt{succ} : \mathbb{N} \to \mathbb{N}$: given $c_0 : C$ and $c_s : \mathbb{N} \to C \to C$, we get $f : \mathbb{N} \to C$ with $f(0) = c_0$ and $f(\mathtt{succ}\ n) = c_s(n, f(n))$.

Note the preference for constructions defined by mapping out.

When the induction principles are generalized to dependent types, uniqueness will follow.

Dependent Types

The above structure is enough to construct types that depend on elements of other types.

These dependent types are one of the key ideas in Martin-Löf type theory, and will play a central role in this talk.

Examples:

$$\lambda a.B : A \longrightarrow \texttt{Type} \quad \text{(a constant type family)}$$
$$\lambda n.A^n : \mathbb{N} \longrightarrow \texttt{Type} \quad (A^{n+1} := A \times A^n, \text{ inductively})$$
$$\lambda(A, B).\, A + B : \texttt{Type} \times \texttt{Type} \longrightarrow \texttt{Type}$$
$$\texttt{parity} : \mathbb{N} \longrightarrow \texttt{Type}$$

with $\texttt{parity}(n) = \varnothing$ for n even and $\texttt{1}$ for n odd.

Dependent Sums and Products

Dependent sums are like the disjoint union:

Given a type family $B : A \to \mathtt{Type}$, the dependent sum $\sum_{a:A} B(a)$ is freely generated by pairs (a, b) with $b : B(a)$.

The dependent sum has a projection map

$$\mathtt{pr}_1 : \sum_{a:A} B(a) \longrightarrow A$$

sending (a, b) to a, and we call such a map a fibration.

There is also a dependent product $\prod_{x:A} B(x)$. Its elements are functions f sending each $x : A$ to an $f(x) : B(x)$.

Note: both the value of $f(x)$ and the type of $f(x)$ depend on x.

$\prod_{a:A} B(a)$ can also be thought of as the space of sections of $\mathtt{pr}_1$.

Propositions as Types: Curry-Howard

A type can be thought of as a proposition, which is true when inhabited:

$$
\begin{array}{rcl}
\text{Types} & \longleftrightarrow & \text{Propositions} \\
\\
\varnothing & \longleftrightarrow & \text{false} \\
\mathtt{1} & \longleftrightarrow & \text{true} \\
P \times Q & \longleftrightarrow & P \text{ and } Q \\
P + Q & \longleftrightarrow & P \text{ or } Q \\
P \to Q & \longleftrightarrow & P \text{ implies } Q \\
\displaystyle\prod_{x:A} P(x) & \longleftrightarrow & \forall x P(x) \\
\displaystyle\sum_{x:A} P(x) & \longleftrightarrow & \exists x P(x)
\end{array}
$$

But what about the proposition $a = b$?

Identity Types

Given a type A, the identity type of A is a type family $A \times A \to \mathtt{Type}$ whose values are written $a = b$ for $a, b : A$.

This type family is generated by "reflexivity" elements of the form $\mathtt{refl}_a : a = a$ for each $a : A$.

An element of the type $a = b$ was historically thought of as the assertion that a equals b, but in the homotopical interpretation should be thought of as a path from a to b in A, with $\mathtt{refl}_a$ being the constant path at a.

The associated map $\sum_{a,b:A} (a = b) \longrightarrow A \times A$ was historically thought of as the diagonal map $A \to A \times A$, but for our purposes it is better regarded as the path fibration $A^I \to A \times A$, which is obtained by replacing the diagonal map by a fibration.

Regular Mathematics

With the foundation presented so far, all of the usual constructions of mathematics can be done, with types thought of as sets.

For example, one can construct the real numbers and do analysis; one can prove theorems in algebra; and one can define topological spaces and simplicial sets, and prove the standard results about them.

Major results include the Feit-Thompson odd-order theorem (Gonthier), the four-colour theorem (Gonthier), Kepler's sphere packing conjecture (Hales), a C compiler that has been proven correct and is used in industry (Leroy et al), and lots more.

Models

A model of type theory is a category equipped with type constructors that satisfy all of the properties we have assumed. (Making this precise is technical.)

$$
\begin{array}{rcl}
\varnothing & \longleftrightarrow & \text{initial object} \\
\mathtt{1} & \longleftrightarrow & \text{terminal object} \\
P \times Q & \longleftrightarrow & \text{product} \\
P + Q & \longleftrightarrow & \text{coproduct} \\
P \to Q & \longleftrightarrow & \text{cartesian closed} \\
\displaystyle\prod_{x:A} P(x) & \longleftrightarrow & \text{locally cartesian closed} \\
a = b & \longleftrightarrow & \text{a weak factorization system}
\end{array}
$$

with suitable compatibility.

Motivating example: Set with epi-mono weak factorization system, so $a = b$ is usual equality.

Models, II

For $a, b : A$, we have a type $a = b$. Therefore, *it* has an associated identity type $p = q$ for $p, q : a = b$. For over 20 years, it was an open question whether $p = q$ always holds.

In 1998, Hofmann and Streicher showed that the category of groupoids is a model of type theory, with $a = b$ given by $\mathrm{Hom}(a, b)$. It follows that the answer is no!

Then in 2006, Voevodsky showed that simplicial sets form a model of type theory, which also shows that the answer is no.

At about the same time, Awodey and Warren showed that weak factorization systems give identity types.

We now know that many Quillen model categories are models of type theory. Homotopical thinking clarifies aspects of type theory.

Any proof in type theory gives a theorem in all models!

Equivalences

The models above suggest thinking of a type as a homotopical object. Let's see where this leads.

We say that $f : A \to B$ is an equivalence if it has left and right inverses. That is,

$$\mathtt{IsEquiv}f \;:\equiv\; \left(\sum_{g:B\to A} (gf = \mathtt{id}_A)\right) \times \left(\sum_{h:B\to A} (fh = \mathtt{id}_B)\right).$$

The type of equivalences from A to B is

$$A \simeq B \;:\equiv\; \sum_{f:A\to B} \mathtt{IsEquiv}f.$$

One can also define loop spaces, homotopy groups, etc.

Univalence Axiom

For types A and B, we define a function $\omega : (A = B) \to (A \simeq B)$ by sending $\mathtt{refl}_A$ to $\mathtt{id}_A$.

The Univalence Axiom says that ω is an equivalence for all types A and B.

If ω is an equivalence, then there is an inverse map

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

which implies that equivalent types are equal.

This is an assertion about the universe `Type`, and it does not hold in the standard set-theoretic model.

But it does hold for the model in simplicial sets. Type theory with this axiom is called Homotopy Type Theory.

With this axiom, one can prove $\pi_1(S^1) = \mathbb{Z}$, $\pi_4(S^2) = \mathbb{Z}/2$, Blakers-Massey, and many other results.

Univalent type families

Univalence is a puzzling notion, so we'll study it by generalizing it.

We say that a type family $B : A \to \mathtt{Type}$ is univalent if the composite map

$$(a = a') \xrightarrow{\mathtt{ap}\ B} (B(a) = B(a')) \xrightarrow{\omega} (B(a) \simeq B(a'))$$

is an equivalence, where $\mathtt{ap}\ B$ sends $\mathtt{refl}_a$ to $\mathtt{refl}_{B(a)}$.

The Univalence Axiom is the special case where the type family is $\mathtt{id}_{\mathtt{Type}} : \mathtt{Type} \to \mathtt{Type}$.

Our goal is to characterize the univalent type families, as a way to better understand the Univalence Axiom.

The Case of Simplicial Sets

In the simplicial model, the notion of equivalence matches the standard notion, and a type family corresponds to a Kan fibration, obtained using the dependent sum construction.

Associated to a Kan fibration $E \to B$ is a fibration $Eq(E) \to B \times B$ whose fibre over (b, b') is the simplicial set of equivalences from the fibre E_b to the fibre $E_{b'}$.

The fibration $E \to B$ is univalent iff a certain natural map from the path space B^I to $Eq(E)$ is an equivalence. (Recall the first slide.)

Examples:

- The identity map $X \to X$ is univalent iff X is empty or contractible.
- The double-cover $S^\infty \to \mathbb{R}P^\infty$ is univalent.

$\mathtt{BAut}(F)$

We next construct a univalent type family associated to any type F. Define

$$\mathtt{BAut}(F) :\equiv \sum_{Z:\mathtt{Type}} |F \simeq Z|,$$

where $|P|$ denotes the propositional truncation of the type P. This is a type which is either empty or contractible, and has the same truth value as P.

Define

$$\mathtt{Aut}(F) :\equiv (F \simeq F),$$

the monoid of self-equivalences of F.

Lemma (Easy)**.** $\Omega\mathtt{BAut}(F) \simeq \mathtt{Aut}(F)$, where $\Omega X :\equiv (x_0 = x_0)$.

I don't know how to define BG in general, but the above implies that `BAut` gives the correct result in a model.

The universal fibration with fibre F

Since $\mathtt{BAut}(F) :\equiv \sum_{Z:\mathtt{Type}} |F \simeq Z|$, we have a canonical map

$$\alpha :\equiv \mathtt{pr}_1 : \mathtt{BAut}(F) \longrightarrow \mathtt{Type}.$$

Proposition. The type family α is univalent for any type F.

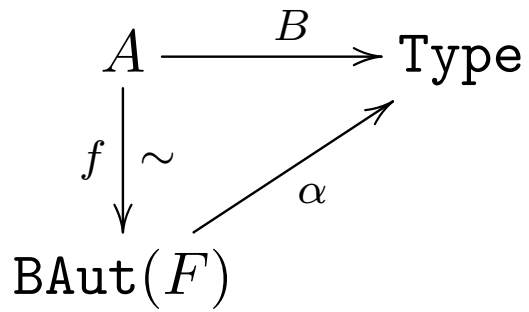
A technical argument was sketched for simplicial sets in a talk by Moerdijk. It is proved for ∞-topoi by Gepner and J. Kock. In fact, the proposition is easy to prove in type theory (see the HoTT library), and implies the result in simplicial sets and other models.

Consider the associated fibration $\sum_{(Z,e):\mathtt{BAut}(F)} Z \to \mathtt{BAut}(F)$.

Taking $F = \mathtt{1}$ gives the identity map on a contractible space.
Taking $F = \mathtt{1} + \mathtt{1}$ gives $S^\infty \to \mathbb{R}P^\infty$. In general, get $\mathrm{EAut}(F) \times_{\mathrm{Aut}(F)} F \to \mathrm{BAut}(F)$.

A Characterization of Univalent Fibrations

Theorem (C)**.** If $B : A \to \mathtt{Type}$ is a univalent type family and A is connected, then there is a type F and an equivalence $f : A \simeq \mathtt{BAut}(F)$ such that



commutes up to homotopy.

More generally, if A is not connected, then it is a coproduct of a set-indexed family of $\mathtt{BAut}(F)$'s, with pairwise non-equivalent F's, with a similar commuting diagram.

The case of the empty coproduct gives $\varnothing \to \varnothing$, our other example.

Consequences

- Univalent fibrations are exactly the classifying bundles for fibrations with fibre F (or an appropriate coproduct of such). This is a notion from the 1950's that is extremely well-studied in topology (Stasheff, May and others).
- If the Univalence Axiom holds, then $\mathtt{id}_{\mathtt{Type}} : \mathtt{Type} \to \mathtt{Type}$ is a univalent type family, so $\mathtt{Type}$ is equivalent to $\sum \mathtt{BAut}(F)$, where the sum is over equivalence classes of $F : \mathtt{Type}$.
  We can also describe the total space of the universal fibration over $\mathtt{Type}$.
- A given type A is rarely the base of a univalent fibration, and when it is, it is usually the base of only one univalent fibration up to equivalence. However, coincidences can occur. E.g. taking F to be $\mathtt{1}$ gives $\mathtt{1} \to \mathtt{1}$, but taking F to be $\varnothing$ gives $\varnothing \to \mathtt{1}$, so both are univalent.

  I know of no other coincidences!

About the Proof

It's almost tautologous that a univalent family $B : A \to \texttt{Type}$ factors through $\texttt{BAut}(F)$ when B is connected. But there is some work in showing that the map $A \to \texttt{BAut}(F)$ is an equivalence.

The proof has been formalized in the proof assistant `Coq`.

I also have a more complicated proof for simplicial sets, that doesn't use a universe or univalence. This could possibly give another way to prove that simplicial sets has a univalent universe, but there are non-trivial issues of strictness.

To learn more about homotopy type theory: These slides and a longer introduction to type theory are on my web site.

Mike Shulman's slides from two series of lectures are great.

Homotopy Type Theory: Univalent Foundations of Mathematics is the standard source.

Thanks!