Synthetic homotopy theory and higher inductive types

Ulrik Buchholtz, CMU

FoMUS, Bielefeld, July, 2016
1 Synthetic Homotopy Theory
2 Truncation
3 The fundamental group of the circle
4 Higher inductive types
5 More homotopy theory
6 More HITs
1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
As a first approximation consider the analogy:

synthetic geometry : analytic geometry
synthetic homotopy theory : classical homotopy theory
Synthetic homotopy theory

As a first approximation consider the analogy:

\[
\begin{align*}
\text{synthetic geometry} & : \quad \text{analytic geometry} \\
\text{synthetic homotopy theory} & : \quad \text{classical homotopy theory}
\end{align*}
\]

Spaces, points, paths, homotopies are *basic notions* given directly in terms of the identity type. Sometimes this leads to *new proofs*. 
Continuity

It’s not a new idea to consider abstract homotopy theory: this goes back at least 50 years:

- Edgar Brown’s abstract homotopy theory (1965)
- Quillen model categories (1967)
- Kenneth Brown’s fibration categories (1973)
- Waldhausen categories (1983)
- Grothendieck’s derivators (1990)
- etc.

In this way, HoTT is part of a long tradition in homotopy theory.

Homotopy type theory provides another way to do abstract homotopy theory. It feels even more synthetic because the framework ensures that everything is invariant under equivalence. This is new.
Some have complained about the term *synthetic homotopy theory* for this reason. Perhaps better would be *type-theoretic homotopy theory* or *univalent homotopy theory*.

What we’ll do in this workshop is to see how this works for a few basic results (kind of like browsing through Book 1 of Euclid’s *Elements*), and along the way we’ll be acquainted with a key tool: higher inductive types.
1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
Let us recall the basic dictionary of homotopy type theory:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Type</th>
<th>Space</th>
<th>Infinite Groupoid</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>type</td>
<td>space</td>
<td>$\infty$-groupoid</td>
</tr>
<tr>
<td>$a : A$</td>
<td>term</td>
<td>point</td>
<td>object</td>
</tr>
<tr>
<td>$p : a =_A b$</td>
<td>identification</td>
<td>path</td>
<td>arrow</td>
</tr>
<tr>
<td>$f : A \to B$</td>
<td>function</td>
<td>continuous map</td>
<td>homomorphism</td>
</tr>
<tr>
<td>$C : A \to Type$</td>
<td>dep. type</td>
<td>fibration</td>
<td>fibrations</td>
</tr>
<tr>
<td>$f : \Pi(x : A)C(x)$</td>
<td>dep. function</td>
<td>section</td>
<td>section</td>
</tr>
<tr>
<td>$s : \Sigma(x : A)C(x)$</td>
<td>dep. pair</td>
<td>point in total space</td>
<td></td>
</tr>
</tbody>
</table>
Recall that a type \( A \) may be *truncated* at a finite level:

<table>
<thead>
<tr>
<th>Level</th>
<th>Predicate</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-2)</td>
<td>( \text{isContr}(A) := \Sigma(x : A)\Pi(y : A)(x = y) )</td>
<td>contractible</td>
</tr>
<tr>
<td>(-1)</td>
<td>( \text{isProp}(A) := \Pi(x, y : A)\text{isContr}(x = y) )</td>
<td>proposition</td>
</tr>
<tr>
<td>0</td>
<td>( \text{isSet}(A) := \Pi(x, y : A)\text{isProp}(x = y) )</td>
<td>set</td>
</tr>
<tr>
<td>1</td>
<td>( \text{isGpd}(A) := \Pi(x, y : A)\text{isSet}(x = y) )</td>
<td>groupoid</td>
</tr>
<tr>
<td>(n+1)</td>
<td>( \text{isTrunc}_{n+1}(A) := \Pi(x, y : A)\text{isTrunc}_n(x = y) )</td>
<td>(n + 1)-groupoid</td>
</tr>
</tbody>
</table>

These predicates are themselves propositions and we have the equivalence

\[
\text{isProp}(A) \simeq \Pi(x, y : A)(x = y)
\]
Recall that the identity types $a = b$, for $a, b : A$ can be thought of as inductively defined by an element $\text{idp} : a = a$.

The corresponding induction principle is called *path induction*: If $C : \Pi(x : A)((a = x) \rightarrow \text{Type})$, and we have some $c : C(a, \text{idp})$, then we have a section

$$J(C, c) : \Pi(x : A)(p : a = x). C(x, p)$$

We have $J(C, c)(a, \text{idp}) = c$. 
A fundamental example of a fibration is the *path fibration*: Given a type $A$ and a point $a : A$, have $P : A \to \text{Type}$ with $P(x) := (a = x)$.

Exercise: prove $\text{isContr}(\Sigma(x : A)P(x))$. 
A fundamental example of a fibration is the *path fibration*: Given a type $A$ and a point $a : A$, have $P : A \to \text{Type}$ with $P(x) := (a = x)$.

Exercise: prove $\text{isContr}(\Sigma(x : A)P(x))$.

Center of contraction: $\langle a, \text{idp} \rangle$.

Use path induction to show $\langle a, \text{idp} \rangle = \langle b, p \rangle$ for any $b : A$, $p : a = b$. 
We need to know that there is a way for any $A$ and $n \geq -2$ to make a “best approximation” $\|A\|_n$ of $A$ that is an $n$-truncated type. It comes with a map $|-|_n : A \to \|A\|_n$.

The universal property of the truncation is this: If $B$ is any $n$-truncated type, then the following map is an equivalence:

$$(\|A\|_n \to B) \to (A \to B)$$

$g \mapsto g \circ |-|_n$

Later we’ll see how to construct $\|A\|_n$ as a higher inductive type.
1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
The fundamental group of a pointed space $A$, written $\pi_1(A, \star)$, has as underlying set

$$\pi_1(A, \star) := \|\Omega(A, \star)\|_0,$$

where $\Omega(A, \star) := (\star =_A \star)$ is the loop space.

The group operation is path concatenation, inverses are given by path reversal, and the neutral element is the reflexivity path.

We’re going to calculate the fundamental group of the circle $S^1$. 

We introduce the circle $S^1$ as motivating example of a higher inductive type. Recall two examples of ordinary inductive types:

The booleans $\mathbb{B}$, generated by:
- Two points $\text{true}, \text{false} : \mathbb{B}$.

The natural numbers $\mathbb{N}$, generated by:
- A point $0 : \mathbb{N}$, and
- a function $S : \mathbb{N} \rightarrow \mathbb{N}$.

The circle is generated by:
- A point $\text{base} : S^1$, and
- A path $\text{loop} : \text{base} =_{S^1} \text{base}$. 
Let’s see what the elimination principle for the circle should be:

- We define \( f : \mathbb{B} \to \mathbb{Z} \) by recursion by giving \( f(\text{true}) : \mathbb{Z} \) and \( f(\text{false}) : \mathbb{Z} \).
- We define \( f : \mathbb{N} \to \mathbb{Z} \) by recursion by giving \( f(0) : \mathbb{Z} \) and for each \( n : \mathbb{N}, f(S\ n) : \mathbb{Z} \), assuming the value \( f(n) : \mathbb{Z} \) known.
- We define \( f : S^1 \to \mathbb{Z} \) by recursion by giving \( f(\text{base}) : \mathbb{Z} \) and \( \text{ap}_f(\text{loop}) \), a path of type \( f(\text{base}) =_\mathbb{Z} f(\text{base}) \).
Induction principles not only tell us how to construction functions out of an inductive type, they also tell us more generally how to construct sections of fibrations over them.

- We define a section $f : \Pi (b : \mathbb{B}) P(b)$ by induction by giving $f(\text{true}) : P(\text{true})$ and $f(\text{false}) : P(\text{false})$.

- We define $f : \Pi (n : \mathbb{N}) P(n)$ by induction by giving $f(0) : P(0)$ and for each $n : \mathbb{N}$, $f(S\ n) : P(S\ n)$, assuming the value $f(n) : P(n)$ known.

- We define $f : \Pi (x : S^1) P(x)$ by induction by giving $f(\text{base}) : P(\text{base})$ and $\text{apd}_f(\text{loop})$, a dependent path (a pathover) of type $f(\text{base}) =^P_{\text{loop}} f(\text{base})$. 
We now prove that $\pi_1(S^1) \simeq \mathbb{Z}$. Define $\text{code} : S^1 \to \text{Type}$ by recursion:

\[
\begin{align*}
\text{code}(\text{base}) & := \mathbb{Z} \\
\text{ap}_{\text{code}}(\text{loop}) & := \text{ua}(\text{succ})
\end{align*}
\]

(Note the use of univalence!)

We then give a fiber-wise equivalence

\[
\Pi(x : S^1)(\text{code}(x) \simeq (\text{base} = x)).
\]

It’s possible to emulate the traditional proof in HoTT, proving that a fiber-wise map gives an equivalence on total spaces:

\[
\Sigma(x : S^1)\text{code}(x) \simeq \Sigma(x : S^1)(\text{base} = x) \simeq 1.
\]
The encode-decode method

Here is a more type-theoretic proof: First define encode : Π(x : S¹)((base = x) → code(x)) by

\[
\text{encode}(x) := \lambda p : \text{base} = x. \ \text{transport}^{\text{code}}(p, 0)
\]

and decode : Π(x : S¹)(code(x) → (base = x)) by circle induction:

\[
\text{decode}(\text{base}) := \lambda z : \mathbb{Z}. \ \text{loop}^z
\]
\[
\text{apd}_{\text{decode}}(\text{loop}) := ?
\]
The encode-decode method

Here is a more type-theoretic proof: First define
encode : \( \Pi(x : S^1)((\text{base} = x) \to \text{code}(x)) \) by

\[
\text{encode}(x) := \lambda p : \text{base} = x. \text{transport}^{\text{code}}(p, 0)
\]

and decode : \( \Pi(x : S^1)(\text{code}(x) \to (\text{base} = x)) \) by circle induction:

\[
\text{decode}(:) := \lambda z : \mathbb{Z}. \text{loop}^z \\
\text{apd}_{\text{decode}}(\text{loop}) := ?
\]

Need lemma: if \( B, C : A \to \text{Type} \), \( p : a =_A a' \), \( f : B a \to C a \),
\( g : B a' \to C a' \) and \( \Pi(b : B a)(f b =_C g (~\text{transport}^B(p, b))) \), then
\( f =_p \lambda x : A. B x \to C x \ g. \)
The encode-decode method, cont.

Need to show for all \( z : \mathbb{Z} \): \( \text{loop}^z = \text{loop}^{\text{base}=x \text{ loop}^{z+1}} \).

Need another lemma: if \( f, g : A \rightarrow B, p : a =_A a', q : f a = g a, \) \( r : f a' = g a' \) and \( s \) fills a square:

\[
\begin{array}{c}
  f a \\| & q&| \quad \| & ap_f p \\
  \quad ap_p p & \quad & \quad \quad \quad & \quad ap_g p \\
  f a' \\| & r&| \quad \| & \quad \\
  \quad g a' & \\
\end{array}
\]

then \( q =_{p} \lambda x : A. f x =_g x \quad r. \)
The encode-decode method, cont.

Still need to show for all $z : \mathbb{Z}$: $\text{loop}^z = \lambda x : S^1. \text{base} = x \text{ loop}^{z+1}$. By the lemma, it suffices to fill the square:

\[
\begin{array}{c}
\text{base} \\
\text{idp} \\
\text{base}
\end{array}
\begin{array}{c}
\text{loop}^z \\
\text{loop} \\
\text{loop}^{z+1}
\end{array}
\text{base}
\]

This we can easily do. Thus we can define the function decode : $\Pi(x : S^1)(\text{code}(x) \rightarrow (\text{base} = x))$.

It remains to show that encode and decode are fiberwise mutually inverse.
Lemma 1: For all $x : S^1$ and $p : \text{base} = x$, 
$\text{decode}(x)(\text{encode}(x)(p)) = p$. 
Proof by path induction: $\text{decode}(\text{base})(\text{encode}(\text{base})(\text{idp})) =$ 
$\text{decode}(\text{base})(0) = \text{loop}^0 = \text{idp}$. □

Lemma 2: For all $x : S^1$ and $z : \text{code}(x)$, $\text{encode}(x)(\text{decode}(x)(z)) = z$. 
Proof by circle induction: Suffices (since $\mathcal{Z}$ is a set) to do the base case: 
$\text{encode}(\text{base})(\text{decode}(\text{base})(z)) =$ 
$\text{transport}^{\text{code}}(\text{loop}^z, 0) = z$ (by induction on $z : \mathcal{Z}$). □.

This completes the proof that for $x : S^1$, $\text{code}(x) \simeq (\text{base} = x)$. In particular, $\mathcal{Z} \simeq (\text{base} = \text{base})$. Hence also $\mathcal{Z} \simeq \pi_1(S^1)$. □
1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
Higher inductive types (HITs)

Having seen an example of a higher inductive type and how it used in synthetic homotopy theory, let us look at more higher inductive types, and make some general remarks.

- A higher inductive type includes ordinary point constructors, but also \textit{path constructors} (with given source and target), and possibly \textit{higher path constructors}.
- Path constructors give \textit{new} elements of identity types (just like univalence does).
- The resulting type is a freely generated $\infty$-groupoid. But by including higher path constructors we can impose “relations” (will return to this).
- If a path constructor has an argument of type $A$, then $n$-paths in $A$ give rise to $n + 1$-paths in the generated type.
- \textit{I should note that we don’t have a general schema for HITs} yet – \textit{but we’re making progress}
Given \( f : C \to A \) and \( g : C \to B \) (forming a span), the pushout is a type \( D \) fitting into a diagram:

\[
\begin{array}{c}
C \xrightarrow{g} B \\
\downarrow f \downarrow \inr \downarrow \inl \\
A \longrightarrow \longrightarrow D
\end{array}
\]

It has point constructors \( \inl \) and \( \inr \), and a path constructor

\[
\text{glue} : \Pi(x : C)(\inl(f x) = \inr(g x)).
\]

The non-dependent elimination principle is simply the universal property.
Let \( D \) be the pushout of the span consisting of \( f : C \to A \) and \( g : C \to B \). Let \( P : D \to \text{Type} \) be given. We can define a section \( s : \Pi(x : D)P(x) \) by giving:

\[
\begin{align*}
    s(\text{inl} \ a) & : P(\text{inl} \ a) & \text{for } a : A \\
    s(\text{inr} \ b) & : P(\text{inl} \ b) & \text{for } b : B \\
    \text{apd}_s(\text{glue}(x)) : s(\text{inl}(f \ x)) =^P_{\text{glue}(x)} s(\text{inr}(g \ x)) & \text{for } x : C
\end{align*}
\]
A glimpse of the menagerie

Coequalizer: \( Q, \text{ given } f, g : A \rightrightarrows B \): point constructor \( q : B \to Q \);
path constructor \( r : \Pi(x : A)(q(f(x)) = q(g(x))) \).

Interval \( \mathbb{I} \): point constructors \( 0, 1 : \mathbb{I} \);
path constructor \( \text{seg} : 0 = 1 \).

Suspension \( \text{susp}(A) \): point constructors \( N, S : \text{susp}(A) \);
path constructor \( \text{merid} : A \to N = S \).

Join \( A \ast B \): point constructors \( \text{inl} : A \to A \ast B, \text{inr} B \to A \ast B \)
path constructor \( \text{glue} : \Pi(a : A)(b : B). \text{inl} a = \text{inr} b \).

Torus \( T^2 \): point constructor \( \text{base} : T^2 \);
path constructors \( p, q : \text{base} = \text{base} \);
2-path constructor: \( s : p \cdot q = q \cdot p \)

These already suffice to do a lot of homotopy theory. They are all definable using pushouts (and standard type operations).
A glimpse of the menagerie

Coequalizer $Q$, given $f, g : A \to B$: point constructor $q : B \to Q$; path constructor $r : \Pi(x : A)(q(f(x)) = q(g(x)))$.

Interval $\mathbb{I}$: point constructors $0, 1 : \mathbb{I}$; path constructor $\text{seg} : 0 = 1$.

Suspension $\text{susp}(A)$: point constructors $N, S : \text{susp}(A)$; path constructor $\text{merid} : A \to N = S$.

Join $A * B$: point constructors $\text{inl} : A \to A * B$, $\text{inr} B \to A * B$ path constructor $\text{glue} : \Pi(a : A)(b : B). \text{inl} a = \text{inr} b$.

Torus $T^2$: point constructor $\text{base} : T^2$; path constructors $p, q : \text{base} = \text{base}$; 2-path constructor: $s : p \cdot q = q \cdot p$

These already suffice to do a lot of homotopy theory. They are all definable using pushouts (and standard type operations).

Exercise: Prove that $\mathbb{I}$ is contractible.
The torus used a 2-path constructor. It turns out that higher path constructors can always be avoided via the hubs-and-spokes method. For $T^2$, instead of the 2-path constructor we could add another point constructor $h : T^2$ (the hub) and a path constructor $s : \Pi(x : S^1)(f x = h)$, where $f : S^1 \to T^2$ is defined by circle-induction, mapping base to base and loop to $p \cdot q \cdot p^{-1} \cdot q^{-1}$. (Drawing on blackboard)

This is exactly the same principle as when we glue in a higher cell along a map $f$ using a pushout:
We return to constructing the truncations $\|A\|_n$ for $n \geq -2$.
Let $\|A\|_{-2} := 1$.

The *propositional truncation* $\|A\| = \|A\|_{-1}$ is the higher inductive type with: point constructor $|-| : A \to \|A\|$; and path constructor $p : \Pi(x, y : \|A\|)(x = y)$.

$\|A\|$ is freely generated by a function from $A$ and the fact that it should be a proposition.

This is our first example of a *recursive* HIT. The universal property follows from the recursion principle (exercise!).
Fact: $A$ is $n$-truncated iff $\Omega^{n+1}(A, a) \ (\simeq \text{Map}_*(S^{n+1}, (A, a)))$ is contractible for all $a : A$.

This suggests the following description of $\|A\|_n$ as a HIT, generated by

- a function $|-| : A \to \|A\|_n$; and
- for each $r : S^{n+1} \to \|A\|_n$, a hub point $h(r) : \|A\|_n$; and
- for each $r : S^{n+1} \to \|A\|_n$, and each $x : S^{n+1}$, a spoke $s_r(x) : h(x) = r(x)$.

Fact: This gives the right universal property (cf. HoTT book).
It turns out that the truncations are definable in terms of pushouts!

- For propositional truncation, this is due to Floris van Doorn and Nicolai Kraus.
- For higher truncations, this is due to Egbert Rijke.
Suppose $A : \text{Set}$ and $R : A \to A \to \text{Prop}$. Then we can form the quotient $A/R$ as the set-coequalizer of the two projections

\[(*) \quad \Sigma(a, b : A) R(a, b) \Rightarrow A.\]

(This is the set-truncation of the type-coequalizer.)

In fact it can be useful in the general case of $A : \text{Type}$ and $R : A \to A \to \text{Type}$ to form the coequalizer $(*)$, as a type-quotient. This is a built-in HIT in the Lean proof assistant, generated by

- a function $q : A \to A/R$;
- for each $a, b : A$ and each $r : R(a, b)$ a path $e(r) : q(a) = q(b)$.  


Higher inductive types can also be used to construct free algebras. For instance, if $A : \text{Set}$, we can construct the free group on $A$, $F(A)$, as generated by:

- A function $\eta : A \rightarrow F(A)$;
- A function $m : F(A) \times F(A) \rightarrow F(A)$;
- An element $e : F(A)$;
- A function $i : F(A) \rightarrow F(A)$;
- For each $x, y, z : F(A)$ a path $m(x, m(y, z)) = m(m(x, y), z)$;
- For each $x : F(A)$ paths $m(x, e) = x = m(e, x)$;
- For each $x : F(A)$ paths $m(x, i(x)) = e = m(i(x), x)$;
- The 0-truncation constructor.
Outline

1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
Higher homotopy groups

We can define the higher homotopy groups of a pointed type $A$ as
\[ \pi_n(A, \star) := \| \Omega^n(A, \star) \|_0 = \pi_1(\Omega^{n-1}(A, \star)) \]. For $n \geq 2$, this is an abelian group (Eckmann-Hilton argument).

<table>
<thead>
<tr>
<th></th>
<th>$S^0$</th>
<th>$S^1$</th>
<th>$S^2$</th>
<th>$S^3$</th>
<th>$S^4$</th>
<th>$S^5$</th>
<th>$S^6$</th>
<th>$S^7$</th>
<th>$S^8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi_1$</td>
<td>0</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_2$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_3$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}$</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_4$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_5$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_6$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}_{12}$</td>
<td>$\mathbb{Z}_{12}$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_7$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z} \times \mathbb{Z}_{12}$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}$</td>
<td>0</td>
</tr>
<tr>
<td>$\pi_8$</td>
<td>0</td>
<td>0</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}^2$</td>
<td>$\mathbb{Z}_{24}$</td>
<td>$\mathbb{Z}_2$</td>
<td>$\mathbb{Z}$</td>
</tr>
</tbody>
</table>
A type $A$ may be more or less connected:

$$\text{isConn}_n(A) := \text{isContr}(\|A\|_n).$$

$-1$-connected=inhabited, $0$-connected=connected, $1$-connected=simply-connected.

We can prove that suspension increases connectivity:

$$\text{isConn}_n(A) \rightarrow \text{isConn}_{n+1}(\text{susp} \ A)$$

Thus, the $n$-sphere $S^n$ is $n - 1$-connected (what is the base case?).
Truncatedness, connectedness and homotopy groups

Lemma 1: If $A$ is $n$-truncated and $a : A$, then $\pi_k(A, a) = 1$ for $k > n$.

Lemma 2: If $A$ is $n$-connected and $a : A$, then $\pi_k(A, a) = 1$ for $k \leq n$.

Corollary: $\pi_k(S^n) = 1$ for $k < n$. 
Recall that we can think of types as $\infty$-groupoids. A pointed, connected type represents an $\infty$-group.

What is the connection between discrete groups and 1-truncated $\infty$-groups?

Answer: Given $BG$, a pointed, connected, 1-truncated type, $G := \Omega BG$ is a discrete group.

Conversely: Given a discrete group $G$, we can construct a pointed, connected, 1-truncated type $BG$ with $G = \Omega BG$ as a HIT!

What are the constructors?

The same procedure produces a univalent category given a set-presented precategory.
In traditional homotopy theory it is relatively complicated to define covering spaces (e.g., using local homeomorphisms).

In HoTT, a covering space of $A$ is simply a map $C : A \rightarrow \text{Set}$, where $\text{Set} := \Sigma(X : \text{Type})\text{isSet}(X)$.

Given $C : A \rightarrow \text{Set}$, we get for any $a : A$ a $\pi_1(A,a)$-set (using that Set is 1-truncated. Favonia showed that you can go back: there is an equivalence between $\pi_1(A,a)$-sets and $A$-covering spaces when $A$ is connected (recovering a classical theorem).
Infinity group actions

Given a pointed connected type $BG$, thought of as an $\infty$-group $G$, an action of $G$ is just a dependent type $X : BG \to \text{Type}$.

The type acted on is the fiber $X(\ast)$, and the action is by transport. The quotient is just the dependent sum $\Sigma(x : BG)X(x)$!

The path fibration $BG \to \text{Type}$, $\lambda x. \ast = x$ corresponds to the right action of $G$ on itself. The quotient is contractible.
The real projective spaces $\mathbb{R}P^n$ are traditionally quotients of $S^n$ by the antipodal map.

Using the HIT $BC_2$ corresponding to the 2-element group $C_2$, note that the iterated joins $M^n : BC_2 \to Type, \lambda x. (\star = x)^{(n+1)}$, are the antipodal actions on the spheres $S^n$, so we can define $\mathbb{R}P^n := \Sigma(x : BC_2)M^n(x)$.

A similar construction defined complex projective spaces.

(This construction is due to Egbert Rijke and myself.)
If $G$ is abelian, we can form higher versions of $BG$, usually called $K(G,n)$. These are $(n-1)$-connected, $n$-truncated types with $\pi_n(K(G,n)) = G$.

$K(G,0) := G$ and $K(G,1) := BG$.

Cohomology is now simply defined as $H^n(A;G) := \|A \to K(G,n)\|_0$. 
1. Synthetic Homotopy Theory
2. Truncation
3. The fundamental group of the circle
4. Higher inductive types
5. More homotopy theory
6. More HITs
The Cauchy reals

In HoTT we can construct the Cauchy-complete reals (without assuming dependent choice), as follows:

\[
\begin{align*}
\text{rat} : \mathbb{Q} & \to \mathbb{R} \\
\text{lim} : (x : \mathbb{Q}_+ \to \mathbb{R}) & \to (\forall \delta, \varepsilon : \mathbb{Q}_+, x_{\delta} \sim_{\delta + \varepsilon} x_{\varepsilon}) \to \mathbb{R} \\
\text{eq} : (u, v : \mathbb{R}) & \to (\forall \varepsilon : \mathbb{Q}_+, u \sim_{\varepsilon} v) \to u =_{\mathbb{R}} v
\end{align*}
\]

eliding clauses for \( \sim_\varepsilon \), and set truncation.

Complicated induction principle!
In HoTT, we can construct the cumulative hierarchy $V$ as a HIT generated by:

\[
\text{set} : (A : \text{Type}) \rightarrow (f : A \rightarrow V) \rightarrow V \\
\text{eq} : (A, B : \text{Type}) \rightarrow (f : A \rightarrow V) \rightarrow (g : B \rightarrow V) \\
\rightarrow (\forall a : A, \exists b : B, f a =_V g b) \rightarrow (\forall b : B, \exists a : A, f a =_V g b) \\
\rightarrow \text{set}(A,f) =_V \text{set}(B,g)
\]

plus a constructor making $V$ 0-truncated.

Again, complicated induction principle!
Where to go from here: HoTT book and . . .

- Licata-Brunerie, $\pi_n(S^n)$ in *Homotopy Type Theory*, 2013.
- Licata-Finster, *Eilenberg-MacLane Spaces in Homotopy Type Theory*, 2014.
- you?