Finitary Higher Inductive Types in the Groupoid Model

Peter Dybjer and Hugo Moeneclaey

Chalmers and ENS Cachan

Workshop on Foundations for the Practical Formalization of Mathematics
Nantes, 26-27 April, 2017
What is a higher inductive type?

1. In ordinary Martin-Löf type theory

   \[ a =_A a' \]

   has one constructor \( \text{refl} : a =_A a \).

2. In Homotopy Type Theory higher inductive types (hits) are types \( A \) where we can have other constructors as well, for all the iterated identity types:

   \[ a =_A a' \]
   \[ p =_{a =_A a'} p' \]
   \[ \theta =_{p =_{a =_A a'} p'} \theta' \]
   \[
   \vdots
   \]

Terminology:

- point constructor for $A$ (level 0)
- path constructor for $a =_A a'$ (level 1)
- surface constructor for $p =_{a =_A a'} p'$ (level 2)
- etc

$n$-hits only have constructors of level $\leq n$. 
1-hits in the HoTT-book

General examples:
- propositional truncation
- pushout

Homotopical examples:
- interval
- circle
- suspension
General examples:
- 0-truncation
- set-quotient

Homotopical examples:
- 2-sphere
- torus

Computer science example:
In this book we do not attempt to give a general formulation of what constitutes a "higher inductive definition" and how to extract the elimination rule from such a definition - indeed, this is a subtle question and the subject of current research. Instead we will rely on some general informal discussion and numerous examples.
Some questions

- What is a good definition of a higher inductive type, that is, what do the types of their constructors look like in general?
- What are their associated elimination and equality rules?
- How do we show the consistency of a general theory of higher inductive types?
- How do we get a "computational interpretation"?
- What is the foundational status of higher inductive types?
- What is their relation to Martin-Löf’s meaning explanations?
- Can we reduce the meaning of higher inductive types to the standard inductive or inductive-recursive types?
Higher-dimensional, univalent type theory

A *reinterpretation* of *intensional* type theory

- type $= \text{weak } \infty$-groupoid (Kan cubical set)
- new rules are validated, e.g., the *univalence* axiom and *higher inductive types*
- constructivity is maintained because Kan cubical set model can be formulated in constructive metathtory (*extensional* type theory) itself justified by Martin-Löf’s (1979) *standard* meaning explanations. Cf work in progress by Bickford and Coquand on an implementation in NuPRL.
Type theory in the groupoid model


- type = groupoid $A = (A_0, A_1, A_2) = (A_0, A_1, =_{A_1(\_\_,\_\_)})$.

- New rules are validated, e.g., univalence axiom in first universe and higher inductive types of level 2.

- Constructivity is maintained because groupoid model can be formulated in constructive metatheory (extensional type theory) itself justified by Martin-Löf’s (1979) standard meaning explanations.
Type theory in the setoid model

A *reinterpretation* of *intensional* type theory

- type = setoid $A = (A_0, A_1) = (A_0, \equiv_A)$.
- new rules are validated, e.g. *higher inductive types* of level 1, including quotient types and algebraic theories $T_{\Sigma,E}$. Cf Basold, Geuvers, van der Weide (2017).
- constructivity is maintained because setoid model can be formulated in constructive metatheory (*extensional* type theory) itself justified by Martin-Löf’s (1979) *standard* meaning explanations.
Let $A$ be a type and $R$ be a binary relation on $A$. Then $A/R$ is the 1-hit with

$$c_0 : A \to A/R$$

$$c_1 : (x, y : A) \to R(x, y) \to c_0(x) =_{A/R} c_0(y)$$

Notation: $[x] = c_0(x)$
Quotient types in the setoid model

In the setoid model the points/elements are generated by the constructor

\[ c_{00} : A_0 \to (A/R)_0 \]

and the paths/proofs of equality are generated by

\[ c_{10} : (x, y : A_0) \to (R(x, y))_0 \to c_{00}(x) =_{A/R} c_{00}(y) \]
\[ c_{01} : (x, y : A_0) \to x =_A y \to c_{00}(x) =_{A/R} c_{00}(y) \]
\[ \circ : (x, y, z \in (A/R)_0) \to x =_{A/R} y \to y =_{A/R} z \to x =_{A/R} z \]
\[ \text{id} : (x \in (A/R)_0) \to x =_{A/R} x \]
\[ (-1) : (x, y \in (A/R)_0) \to x =_{A/R} y \to y =_{A/R} x \]

Note that \((A/R)_0\) is an inductive type and \(=_{A/R}\) is an inductive family which are instances of the general schema for inductive families of Dybjer (1991) and CiC.
Heterogenous identity

- If $x : A \vdash C(x)$, $a, a' : A$, and $p : a =_A a'$, then
  \[ c \overset{C}{=} p c' \]
  denotes the heterogenous identity of $c : C(a)$ and $c' : C(a')$.

- If $f : (x : A) \to C(x)$, $a, a' : A$, then
  \[ \text{apd}_f : (p : a =_A a') \to f(a) \overset{C}{=} p f(a') \]
  Both are definable from the rules for homogeneous identity types.
  (Should they perhaps be primitive?)
The elimination rule expresses how to define a function

\[ f : (x : A/R) \rightarrow C(x) \]

by structural induction on the points of \( A/R \), such that the function preserves \( =_{A/R} \).

\[
\begin{align*}
f(c_0(x)) &= \tilde{c}_0(x) \\
apd_f(c_1(x, y, z)) &= \tilde{c}_1(x, y, z)
\end{align*}
\]

under the assumptions

\[
\begin{align*}
\tilde{c}_0 & : (x : A) \rightarrow C(c_0(x)) \\
\tilde{c}_1 & : (x, y : A) \rightarrow (z : R(x, y)) \rightarrow c_0(x) = c_{c_1(x, y, z)} \tilde{c}_0(y)
\end{align*}
\]
General schema for 1-hits?

$H$ is a hit with point constructors

$c_0 : ?$

and path constructors

$c_1 : ?$

What is the form of their types?
General schema for 1-hits?

\( \mathcal{H} \) is a hit with point constructors

\[ c_0 : ? \]

and path constructors

\[ c_1 : ? \]

What is the form of their types? First try:

- the type of a point constructor has the form of a constructor for an inductive type \( \mathcal{H} \).

- the type of a path constructor has the form of a constructor for a binary inductive family \( \equiv_\mathcal{H} \) on \( \mathcal{H} \).
We settle for the time being for a restricted version of hits:

- the type of a point constructor has the form of a constructor for a *finitary* inductive type $H$.
- the type of a path constructor has the form of a constructor for a *finitary* binary inductive family $=_H$ on $H$. The indices in the type are *point constructor patterns*.

Three reasons:

- Simpler semantics
- Simpler syntax, yet cover most (but not all) examples
- Clearly constructive (the schema for inductive families in Dybjer (1991) was perhaps too general)
The type of a point constructor

Finitely branching trees, with finitely many constructors

\[ c_0 : (x_1 : A_1) \rightarrow \cdots \rightarrow (x_n : A_n(x_1, \ldots, x_{n-1})) \]
\[ \rightarrow H \rightarrow \cdots \rightarrow H \]
\[ \rightarrow H \]

\(A_i\) are arbitrary types. They may not depend on \(H\).
This is also the schema for point constructors of the hit \(H\).
A schema for path constructors

\[ c_1 : (x_1 : A_1) \to \cdots \to (x_n : A_n(x_1, \ldots, x_{n-1})) \]

\[ \to (y_1 : H) \to \cdots \to (y_{n'} : H) \]

\[ \to p_1(x_i, y_j) =_H q_1(x_i, y_j) \to \cdots \to p_m(x_i, y_j) =_H q_m(x_i, y_j) \]

\[ \to p'(x_i, y_j) =_H q'(x_i, y_j) \]

where neither \( H \) nor \( =_H \) may appear in \( A_i \) and where \( p_1, q_1, \ldots, p_m, q_m, p', q' \) are point constructor patterns built up by from variables \( x_i, y_j \) by point constructors \( c_0 \).

- 1-hits generalize \( T_{\Sigma,E} \) from algebraic specification theory, the initial term algebra for a signature \( \Sigma \) and a list of equations \( E \).
- Note that although one may think that the set of points of \( H \) is defined before \( =_H \), a negative occurrence of \( H \) would generate a negative occurrence of \( =_H \) in the setoid interpretation of \( =_H \).
A simplified form with only one side condition and one inductive premise:

\[
\begin{align*}
  c_0 & : A_0 \to H \to H \\
  c_1 & : (x : A_1) \to (y : H) \to p(x, y) =_H q(x, y) \\
  & \to p'(x, y) =_H q'(x, y)
\end{align*}
\]
The Torus $\mathbb{T}^2$ as a 2-hit

\[
\begin{align*}
\text{base} & : \quad \mathbb{T}^2 \\
\text{path}_1 & : \quad \text{base} =_{T^2} \text{base} \\
\text{path}_2 & : \quad \text{base} =_{T^2} \text{base} \\
\text{surf} & : \quad \text{path}_1 \cdot \text{path}_2 =_{\text{base}=_{T^2}\text{base}} \text{path}_2 \cdot \text{path}_1
\end{align*}
\]
Simplified schema for 2-hits

Simplified version:

\[ c_0 : A_0 \rightarrow H \rightarrow H \]
\[ c_1 : (x : A_1) \rightarrow (y : H) \rightarrow p(x, y) =_H q(x, y) \]
\[ \rightarrow p_1(x, y) =_H q_1(x, y) \]
\[ c_2 : (x : A_2) \rightarrow (y : H) \rightarrow (z : p_2(x, y) =_H q_2(x, y)) \]
\[ \rightarrow g_1(x, y, z) =_{p_3(x,y)=_H q_3(x,y)} h_1(x, y, z) \]
\[ \rightarrow g_2(x, y, z) =_{p_4(x,y)=_H q_4(x,y)} h_2(x, y, z) \]

Here \( p, q, p_i, q_i \) are point constructor patterns in the variables \( x, y \)
and \( g_i, h_i \) are path constructor patterns in the variables \( x, y, z \).
Point and path constructor patterns

Point constructor patterns

\[ p ::= x | c_0(a, p) \]

Path constructor patterns

\[ g ::= z | c_1(a, p, g) | g \circ g | id | g^{-1} \]

(add \( \text{ap}_{c_0}(p, g) \)?)
The elimination rule expresses how to define a function

\[ f : (x : H) \rightarrow C(x) \]

by structural induction on the points of \( H \), such that the function preserves \( =_H \).

\[
\begin{align*}
f(c_0(x, y)) &= \tilde{c}_0(x, y, f(y)) \\
\text{apd}_f(c_1(x, y, z)) &= \tilde{c}_1(x, y, f(y), z, \text{apd}_f(z))
\end{align*}
\]

under the assumptions

\[
\begin{align*}
\tilde{c}_0 & : (x : A_0) \rightarrow (y : H) \rightarrow C(y) \rightarrow C(c_0(x, y)) \\
\tilde{c}_1 & : (x : A_1) \rightarrow (y : H) \rightarrow (\tilde{y} : C(y)) \\
& \rightarrow (z : p =_H q) \rightarrow T_0(p) =_Z T_0(q) \rightarrow T_0(p') =_C^{c_1(x, y, z)} T_0(q')
\end{align*}
\]

where \( T_0 \) is a lifting function defined below.
Lifting point constructor patterns:

\[ T_0(y) = \tilde{y} \]
\[ T_0(c_0(a, p)) = \tilde{c}_0(a, p, T_0(p)) \]

Lifting path constructor patterns:

\[ T_1(z) = \tilde{z} \]
\[ T_1(c_1(a, p, g)) = \tilde{c}_1(a, p, T_0(p), g, T_1(g)) \]
\[ T_1(g \circ g') = T_1(g) \circ' T_1(g') \]
\[ T_1(\text{id}) = \text{id} \]
\[ T_1(g^{-1}) = T_1(g)^{-1} \]

It follows that \( T_0(p) = f(p) \) and \( T_1(g) = ap_f(g) \)
Heterogeneous identity of level 2.

Let $a, a' : A$, $p, p' : a =_A a'$, $\theta : p =_{a =_A a'} p'$, $b : B(a)$, $b' : B(a')$, $q : b =_p b'$, $q' : b =_{p'} b'$ We write

$$q =_{\theta} b =_p b'$$

for the heterogeneous identity of the heterogeneous paths $q, q'$. Moreover,

$$q =_{\text{refl}(p)} b =_p b'$$

is judgmentally equal to $q =_{b =_p b'} q'$. 
Functions preserve level 2 identities

If

\[ f : (x : A) \rightarrow C(x) \]

then not only

\[ \text{apd}_f : (p : x =_A x') \rightarrow f(x) =^C_p f(x') \]

but also

\[ \text{apd}^2_f : (\theta : p =_{x=A} x' \; p') \rightarrow \text{apd}_f(p) =_{\theta}^{f(x)=^C f(x')} \text{apd}_f(p') \]
We define $f : (x : H) \to C(x)$ by

\[
\begin{align*}
    f(c_0(a_1, b_1)) &= \tilde{c}_0(a_1, b_1, f(b_1)) \\
    \text{apd}_f(c_1(a_2, b_2, c_2)) &= \tilde{c}_1(a_2, b_2, f(b_2), c_2, \text{apd}_f(c_2)) \\
    \text{apd}^2_f(c_2(a_3, b_3, c_3, d_3)) &= \tilde{c}_2(a_3, b_3, f(b_3), c_3, \text{apd}_f(c_3), d_3, \text{apd}^2_f(d_3))
\end{align*}
\]

We have already shown the assumptions on $\tilde{c}_0$ and $\tilde{c}_1$. We also have

\[
\begin{align*}
    \tilde{c}_2 : & (a_3 : A_2) \to (b_3 : H) \to (\tilde{b}_3 : C(b_3)) \to (c_3 : p_3 =_H q_3) \\
    & \to (\tilde{c}_3 : T_0(p_3) =^C_{c_3} T_0(q_3)) \to (d_3 : g_1 =_{p_4=Hq_4} h_1) \\
    & \to T_1(g_1) =^{T_0(p_4)=_HT_0(q_4)}_{d_3} T_1(h_1) \\
    & \to T_1(g_2) =^{T_0(p_5)=_HT_0(q_5)}_{c_2(a_3,b_3,c_3,d_3)} T_1(h_2)
\end{align*}
\]
The interpretation of $H$ is the groupoid $(H_0, H_1, H_2)$, where

- $H_0$ is the inductively defined set of objects (elements, points).
- $H_1(x, y)$ is the inductively defined family of set of arrows (identity proofs, paths)
- $H_2(x, y, f, g)$ is the inductively defined family of set of 2-cells (identity proofs of arrows, surfaces, homotopies)
The objects of $H$

$H_0$ is inductively generated by a constructor for the object part of the point constructor

\[ c_{00} : (A_0)_0 \rightarrow H_0 \rightarrow H_0 \]
The arrows of $H$

$H_1$ is inductively generated by:

- a constructor for the object part of the path constructor

$$c_{10} : (x \in (A_1)_0) \rightarrow (y \in H_0)$$
$$\rightarrow H_1(p_0(x, y), q_0(x, y)) \rightarrow H_1(p'_0(x, y), q'_0(x, y))$$

- a constructor for the arrow part of the point constructor:

$$c_{01} : (x, x' \in (A_0)_0) \rightarrow (A_0)_1(x, x') \rightarrow (y, y' \in H_0)$$
$$\rightarrow H_1(y, y') \rightarrow H_1(c_{00}(x, y), c_{00}(x', y'))$$

- constructors for composition, identity, and inverse of paths

$$\circ : (x, y, z \in H_0) \rightarrow H_1(x, y) \rightarrow H_1(y, z) \rightarrow H_1(x, z)$$
$$\text{id} : (x \in H_0) \rightarrow H_1(x, x)$$
$$(\cdot)^{-1} : (x, y \in H_0) \rightarrow H_1(x, y) \rightarrow H_1(y, x)$$
The surfaces of $H$

$H_2$ (representing equality of paths) is inductively generated by

- $c_{20}$ – the object part of the surface constructor
- $c_{11}$ – the arrow part of the path constructor
- $c_{02}$ – the surface (preservation of equality of arrows) part of the point constructor:
- $c^{\text{id}}_0, c^\circ_0$ – witnesses for the functor laws for the point constructor
- $\text{tran}, \text{refl}, \text{sym}$ – witnesses that $H_2$ is a family of equivalence relations
- $w_0, w_1$ – witnesses that composition preserves equality
- $\alpha, \lambda, \rho, \iota_0, \iota_1$ – witnesses for the groupoid laws
It’s clear that \((H_0, H_1, H_2)\) is a groupoid.
Interpretation of introduction rules

- The point constructor $c_0 : A_0 \rightarrow H \rightarrow H$ is interpreted by the functor on groupoids with object part $c_{00}$, arrow part $c_{01}$ and preservation of equality part $c_{02}$. The functor laws are witnessed by the constructors $c_0^{\text{id}}$ and $c_0^{\circ}$.

- A groupoid interpreting $x =_H y$ is a setoid and hence functors on such groupoids degenerate to setoid-maps. Hence, the path constructor

$$c_1 : (x : A_1) \rightarrow (y : H) \rightarrow p(x, y) =_H q(x, y)$$
$$\rightarrow p_1(x, y) =_H q_1(x, y)$$

is interpreted by the setoid map with underlying function $c_{10}$ and preservation of equality part $c_{11}$.

- A groupoid interpreting $f =_{x =_H x'} f'$ has only one object and one arrow (up to equality). Hence it suffices that the constructor $c_2$ is interpreted by $c_{20}$. 
Interpretation of elimination and equality rules

We want to show that there exists a "dependent groupoid functor"

\[ f : (x : H) \rightarrow C(x) \]

such that

\[ f(c_0(x, y)) = \tilde{c}_0(x, y, f(y)) \]
\[ \text{apd}_f(c_1(x, y, z)) = \tilde{c}_1(x, y, f(y), z, \text{apd}_f(z)) \]
\[ \text{apd}^2_f(c_2(x, y, z, w)) = \tilde{c}_2(x, y, f(y), z, \text{apd}_f(z), w, \text{apd}^2_f(w)) \]
Object and arrow part of \( f \)

- **Object part** \( f_0 : (x \in H_0) \rightarrow C_0(x) \) by

  \[
  f_0(c_{00}(x, y)) = (\tilde{c}_0)_0(x, y, f_0(y))
  \]

- **Arrow part**

  \[
  f_1 : (x, x' \in H_0) \rightarrow (g \in H_1(x, x')) \rightarrow C'_1(g, f_0(x), f_0(x'))
  \]

  where \( C'_1 \) is a heterogenous version of arrow (between elements of different fibers). This is done by \( H_1 \)-elimination:

  \[
  f_1(c_{10}(x, y, z)) = (\tilde{c}_1)_0(x, y, f_0(y), z, f_1(p, q, z))
  
  f_1(c_{01}(x, x', e, y, y', d)) = (\tilde{c}_0)_1(x, x', e, y, y', d, f_0(y), f_0(y'), f_1(y,
  
  and clauses which say that \( f_1 \) maps an identity on \( H \) to an identity, a composition to a composition, and an inverse to an inverse.
We define the 2-cell part

\[f_2 : (x, x' \in H_0) \rightarrow (g, g' \in H_1(x, x')) \rightarrow (\ast \in H_2(x, x', g, g')) \rightarrow C'_2(\ast, f_1(x, x', g), f_1(x, x', g'))\]

where \(C'_2\) is a heterogenous notion of equality between elements in different fibres. This is proved by \(H_2\)-elimination.
Can the schemata for 1- and 2-hits be extended to arbitrary $n$-hits and also to $\infty$-hits?

- Can cubical type theory be extended with schema for hits with constructors of arbitrary dimensionality?
- Can these hits be interpreted in the Kan cubical set model?

A step on the way:

- Formulate 1- and 2-hits using face maps and degeneracies.
- Formulate setoids and groupoids as truncated Kan cubical sets.