Finitary Higher Inductive Types in the Groupoid Model

Peter Dybjer and Hugo Moeneclaey

Chalmers and ENS Paris-Saclay

Gothenburg, December 12, 2018

Summary

Summary

Two different equalities in dependent type theories

There are the usual *judgmental* equalities (which are decidable).

To be able to use induction we need *propositional* equalities.
Roughly :

- For any type A and $x, y : A$, we have an *identity* type $x =_A y$.
- We have a canonical inhabitant of $x =_A x$.
- If $x =_A y$ is inhabited, then we can substitute x by y.

Extensional type theory

How do these identity types look like ?

Extensional type theories

Any type $x =_A y$ has at most one element.

This rule is not derivable.

Are there meaningful axioms which implies non-trivial identity types ?

Homotopy type theory

It is an extension of dependent type theory.

Two features

- Univalence axiom
- Higher inductive types

Univalence implies non-trivial identity types.

It has a topological interpretation.

Higher inductive types

Intuition

We generate inductively :

- a type H,
- its identity types $x =_{\mathrm{H}} x'$,
- its identity types of identity types $p =_{x =_{\mathrm{H}} x'} p'$,
- etc...

So the type H has constructors building paths, surfaces, ...

Higher inductive types of level n

Terminology:

- point constructors for H (level 0)
- path constructors for $x =_{\mathrm{H}} x'$ (level 1)
- surface constructors for $p =_{x =_{\mathrm{H}} x'} p'$ (level 2)
- etc...

n-hits only have constructors of level $\leq n$.

We deal with 2-hits only.

Examples of 1-hits

Example : Constructors for combinatory logic CL

$$
\begin{aligned}
\mathrm{K} &: \mathrm{CL} \\
\mathrm{S} &: \mathrm{CL} \\
\mathrm{app} &: \mathrm{CL} \to \mathrm{CL} \to \mathrm{CL} \\
\mathrm{K}_{\mathrm{conv}} &: (x, y : \mathrm{CL}) \to \mathrm{app}(\mathrm{app}(\mathrm{K}, x), y) =_{\mathrm{CL}} x \\
\mathrm{S}_{\mathrm{conv}} &: (x, y, z : \mathrm{CL}) \to \mathrm{app}(\mathrm{app}(\mathrm{app}(\mathrm{S}, x), y), z) =_{\mathrm{CL}} \\
&\quad \mathrm{app}(\mathrm{app}(x, z), \mathrm{app}(y, z))
\end{aligned}
$$

Semantically, it is natural to interpret CL as a setoid (i.e. a set with an equivalence relation on it).

Example : Circle S^1

$$\begin{aligned} \text{base} &: S^1 \\ \text{path} &: \text{base} =_{S^1} \text{base} \end{aligned}$$

As a setoid it would be trivial.

Semantically, it is natural to interpret S^1 as some topological object.

Groupoids

Definition

A groupoid is a category where all morphisms are invertible.

How can these be topological objects ?

The fundamental groupoid

To a space X we associate its fundamental groupoid $\pi(X)$:

- objects are the points of X,
- morphisms are path up to continuous deformations.

The fundamental groupoid $\pi(C)$ of the topological circle C is not trivial.
The hit S^1 will be interpreted as (equivalent to) $\pi(C)$.

Plan

We will give a definition for some *finitary* 2-hits and interpret them in the groupoid model of type theory.

Remark

Officially we work in set theory, although we conjecture our work can be done in extensional type theory.

Summary

1. Introduction to higher inductive types
2. Schema for finitary 2-hits
3. Interpretation in the groupoid model

Point constructors for H

Usual constructors for an inductive type T

$$
\begin{aligned}
&(x_1 : A_1) \to \cdots \to (x_m : A_m(x_1, \ldots, x_{m-1})) \\
&\to (B_{1,1}(x_1, \ldots, x_m) \to \cdots \to B_{1,k_1}(x_1, \ldots, x_m) \to \mathrm{T}) \\
&\to \cdots \\
&\to (B_{n,1}(x_1, \ldots, x_m) \to \cdots \to B_{n,k_n}(x_1, \ldots, x_m) \to \mathrm{T}) \\
&\to \mathrm{T}
\end{aligned}
$$

Where T is not occurring in A_i and $B_{j,l}$.

We restrict to finitary hits, i.e. we assume :

Point constructors for a *finitary* hit H

$$
\begin{aligned}
\mathrm{c}_0 \quad : \quad &(x_1 : A_1) \to \cdots \to (x_m : A_m(x_1, \ldots, x_{m-1})) \\
&\to \mathrm{H} \to \cdots \to \mathrm{H} \to \mathrm{H}
\end{aligned}
$$

Path constructors for H

Path constructors for a finitary hit H

$$
\begin{aligned}
\mathrm{c}_1 \quad : \quad & (x_1 : C_1) \to \cdots \to (x_n : C_n(x_1, \ldots, x_n)) \\
& \to (y_1 : \mathrm{H}) \to \cdots \to (y_{n'} : \mathrm{H}) \\
& \to p_1(x_1, \ldots, x_n, y_1, \ldots, y_{n'}) =_{\mathrm{H}} q_1(x_1, \ldots, x_n, y_1, \ldots, y_{n'}) \\
& \vdots \\
& \to p_{n''}(x_1, \ldots, x_n, y_1, \ldots, y_{n'}) =_{\mathrm{H}} q_{n''}(x_1, \ldots, x_n, y_1, \ldots, y_{n'}) \\
& \to p'(x_1, \ldots, x_n, y_1, \ldots, y_{n'}) =_{\mathrm{H}} q'(x_1, \ldots, x_n, y_1, \ldots, y_{n'})
\end{aligned}
$$

Remark :

- H appearing anywhere in C_i contradicts univalence.

A simplified schema

Constructors for a 2-hit H

$$
\begin{aligned}
\mathrm{c}_0 \quad &: \quad A \to \mathrm{H} \to \mathrm{H} \\
\mathrm{c}_1 \quad &: \quad (x : B) \to (y : \mathrm{H}) \to p(x,y) =_{\mathrm{H}} q(x,y) \\
&\quad \to p'(x,y) =_{\mathrm{H}} q'(x,y) \\
\mathrm{c}_2 \quad &: \quad (x : D) \to (y : \mathrm{H}) \to (z : p_3(x,y) =_{\mathrm{H}} q_3(x,y)) \\
&\quad \to g_1(x,y,z) =_{p_4(x,y) =_{\mathrm{H}} q_4(x,y)} h_1(x,y,z) \\
&\quad \to g_2(x,y,z) =_{p_5(x,y) =_{\mathrm{H}} q_5(x,y)} h_2(x,y,z)
\end{aligned}
$$

Where :

- A, B, D are types without H.
- $p, q, p', q', p_3, q_3 \ldots$ are *point constructor patterns*.
- g_1, h_1, g_2, h_2 are *path constructor patterns*

Point and path patterns

Point constructor patterns

$$p ::= y \mid \mathrm{c}_0(a, p)$$

with $y : \mathrm{H}$ and $a : A$ without H.

Path constructor patterns

$$g ::= z \mid \mathrm{c}_1(b, p, g) \mid \mathrm{id} \mid g \circ g \mid g^{-1}$$

with $z : p_3 =_H q_3$ and $b : B$ without H.

Elimination principle

For $x : \mathrm{H} \vdash C(x)$, how can we use induction to define $f : (x : \mathrm{H}) \to C(x)$?

We can define f by pattern matching :

$$\begin{array}{rcl} f(\mathrm{c}_0(x,y)) & = & \tilde{\mathrm{c}_0}(x,y,f(y)) \\ \mathbf{apd}_f(\mathrm{c}_1(x,y,z)) & = & \tilde{\mathrm{c}_1}(x,y,f(y),z,\mathbf{apd}_f(z)) \\ \mathbf{apd}^2_f(\mathrm{c}_2(x,y,z,t)) & = & \tilde{\mathrm{c}_2}(x,y,f(y),z,\mathbf{apd}_f(z),t,\mathbf{apd}^2_f(t)) \end{array}$$

These are **judgmental** equalities.

With suitable $\tilde{c_0}, \tilde{c_1}, \tilde{c_2}$, we can show this schema is well typed **using**

$$\begin{array}{rcl} \mathbf{apd}_f(\mathrm{id}) & = & \mathrm{id} \\ \mathbf{apd}_f(p \circ q) & = & \mathbf{apd}_f(p) \circ' \mathbf{apd}_f(q) \\ \mathbf{apd}_f(p^{-1}) & = & \mathbf{apd}_f(p)^{-1'} \end{array}$$

These equations are valid in the groupoid model.

What are $\tilde{c_0}$, $\tilde{c_1}$ and $\tilde{c_2}$?

We will ask :

$$f(\mathrm{c}_0(x, y)) \quad = \quad \tilde{\mathrm{c}_0}(x, y, f(y))$$

What is $\tilde{c_0}$?

$$\tilde{\mathrm{c}_0} \quad : \quad (x : A) \to (y : \mathrm{H}) \to C(y) \to C(\mathrm{c}_0(x, y))$$

We will ask :

$$\mathbf{apd}_f(\mathrm{c}_1(x,y,z)) \quad = \quad \tilde{\mathrm{c}_1}(x,y,f(y),z,\mathbf{apd}_f(z))$$

What is $\tilde{\mathrm{c}_1}$?

$$\begin{aligned}\tilde{\mathrm{c}_1} \quad : \quad & (x:B) \to (y:\mathrm{H}) \to (\tilde{y}:C(y)) \\ & \to (z : p =_{\mathrm{H}} q) \to \mathrm{T}_0(p) =^C_z \mathrm{T}_0(q) \\ & \to \mathrm{T}_0(p') =^C_{\mathrm{c}_1(x,y,z)} \mathrm{T}_0(q')\end{aligned}$$

$T_0(p)$ is the *lifting* of p (meant to be $f(p)$) defined by :

$$\begin{aligned}\mathrm{T}_0(y) \quad &= \quad \tilde{y} \\ \mathrm{T}_0(\mathrm{c}_0(a,p)) \quad &= \quad \tilde{\mathrm{c}_0}(a,p,\mathrm{T}_0(p))\end{aligned}$$

What is $\tilde{c_2}$?

$$
\begin{aligned}
\tilde{c_2} \quad : \quad & (x : D) \to (y : \mathrm{H}) \to (\tilde{y} : C(y)) \to (z : p_3 =_{\mathrm{H}} q_3) \\
& \to (\tilde{z} : \mathrm{T}_0(p_3) =^{C}_{z} \mathrm{T}_0(q_3)) \to (t : g_1 =_{p_4 =_{\mathrm{H}} q_4} h_1) \\
& \to \mathrm{T}_1(g_1) =_{t}^{\mathrm{T}_0(p_4) =^{\mathrm{H}}_{-} \mathrm{T}_0(q_4)} \mathrm{T}_1(h_1) \\
& \to \mathrm{T}_1(g_2) =_{\mathrm{c}_2(x,y,z,t)}^{\mathrm{T}_0(p_5) =^{\mathrm{H}}_{-} \mathrm{T}_0(q_5)} \mathrm{T}_1(h_2)
\end{aligned}
$$

Where $\mathrm{T}_1(g)$ is the *lifting* of g (meant to be $\mathbf{apd}_f(g)$) defined by :

$$
\begin{aligned}
\mathrm{T}_1(z) &= \tilde{z} \\
\mathrm{T}_1(\mathrm{c}_1(x, y, g)) &= \tilde{c_1}(x, y, \mathrm{T}_0(y), g, \mathrm{T}_1(g)) \\
\mathrm{T}_1(\mathrm{id}) &= \mathrm{id} \\
\mathrm{T}_1(g \circ g') &= \mathrm{T}_1(g) \circ' \mathrm{T}_1(g') \\
\mathrm{T}_1(g^{-1}) &= \mathrm{T}_1(g)^{-1'}
\end{aligned}
$$

Summary

Alternate presentation of groupoids

Definition

A groupoid is a triple :

$$(A_0, A_1, A_2)$$

where

- A_0 is the underlying set.
- For $x, x' \in A_0$, we have $A_1(x, x')$ the set of morphisms between x and x'.
- For $f, f' \in A_1(f, f')$, we have $A_2(f, f')$ inhabited iff $f = f'$.

together with witnesses of the usual groupoid laws.

Groupoid model

We use the groupoid model.
Some correspondences :

$\vdash C$	C is a groupoid
$x : A \vdash C(x)$	C is a functor from A to the category of groupoids
$\vdash f : A \to B$	f is a functor from A to B
$\vdash f : (x : A) \to C(x)$	f is a dependent functor between groupoids

Sketch of the interpretation

Assume H given, we want to show it can be interpreted in the groupoid model.

1. We will build the groupoid $(\mathrm{H}_0, \mathrm{H}_1, \mathrm{H}_2)$ using inductive definition.
2. We do so by building first H_0, then H_1 and finally H_2. We can do this because we deal with finitary hits.
3. Then we check that the introduction, elimination and equality rules are validated by this interpretation.

H_0

Inductively generated by the underlying function of c_0

$$\mathrm{c}_{00} \quad \in \quad A_0 \to \mathrm{H}_0 \to \mathrm{H}_0$$

H_1

Inductively generated by

- The underlying function of c_1

$$\begin{aligned}\mathrm{c}_{10} \quad \in \quad & (x \in B_0) \to (y \in \mathrm{H}_0) \to \mathrm{H}_1(p_0(x,y), q_0(x,y)) \\ & \to \mathrm{H}_1(p_0'(x,y), q_0'(x,y))\end{aligned}$$

- The action of c_0 on paths

$$\begin{aligned}\mathrm{c}_{01} \quad \in \quad & (x, x' \in A_0) \to A_1(x, x') \to (y, y' \in \mathrm{H}_0) \\ & \to \mathrm{H}_1(y, y') \to \mathrm{H}_1(\mathrm{c}_{00}(x,y), \mathrm{c}_{00}(x', y'))\end{aligned}$$

and $\circ, \mathrm{id}, (-)^{-1}$.

H_2

Inductively generated by

- c_{20} – the underlying function of the surface constructor.
- c_{11} – the action on paths of the path constructor.
- c_{02} – the action on surfaces of the point constructor.
- witnesses of the functor laws for the point constructor.
- witnesses of the groupoid laws.

Elimination principle

We need to check that given $x : \mathrm{H} \vdash C(x)$ and suitable constructor $\tilde{c_0}, \tilde{c_1}, \tilde{c_2}$ we are able to build a function $f : (x : \mathrm{H}) \to C(x)$.

1. We build the underlying function f_0 by induction on H_0.
2. We build the action on arrows f_1 by induction on H_1.
3. We show f preserves equalities of paths by building f_2 using induction on H_2.

The judgmental equality for f are immediate from its definition.

Why finitary hits ?

Assume a constructor

$$\mathrm{c}_0 : (A \to \mathrm{H}) \to \mathrm{H}$$

Then H_0 should have a constructor like

$$\begin{aligned} \mathrm{c}_{00} \quad \in \quad & (f_0 \in A_0 \to \mathrm{H}_0) \\ & \to (f_1 \in (a, b \in A_0) \to A_1(a, b) \to \mathrm{H}_1(f_0(a), f_0(b))) \\ & \to \cdots \\ & \to \mathrm{H}_0 \end{aligned}$$

So H_0 and H_1 are generated **at the same time**.

Further work

- This work should be implemented in some proof assistant :
  - We should prove the schema is well-typed.
  - We should prove the groupoid model is correct.
- It is probably possible to extend this method to *infinitary* hits, perhaps using inductive-inductive definition in the model.
- How can point and path constructor patterns be generalised ?
- Can this method be extended to n-hits for arbitrary n ?
- Can this method be extended to ∞-hits, using e.g. Kan cubical sets ?
- Are finitary higher inductive types consistent relatively to inductive families ?