Induction and homotopy initiality for a class of 1-HITs

Gabe Dijkstra

University of Nottingham

May 20th, 2016

j.w.w. Thorsten Altenkirch, Paolo Capriotti and Fredrik Nordvall Forsberg

Aims

Ultimately, we want to have a formal definition of higher inductive types in type theory, such that we can do things like:

- show that initiality and induction coincide
- prove correctness of hub-spokes construction
- show that we get all higher inductive types from some primitive higher inductive type

Aim of this talk

In this talk, we will show:

- how to show that induction and initiality coincide generally
- how to specify 1-HITs in type theory
- what coherence issues show up when we do so
- how to deal with these by restricting ourselves to a smaller class of 1-HITs
- how to derive an induction principle for these 1-HITs

Category theory in type theory

- We want to do category theory in type theory
- We do not want to truncate anything: we work with hom-*types*, not hom-*sets*
- We also do not want to talk about $(\infty, 1)$-categories
- Instead, we deal with coherence *lazily*: we keep track of how much structure and coherence we need from our categories and functors and provide exactly that

Initiality and induction

Given an endofunctor $F : \mathrm{Type} \to \mathrm{Type}$, we can think of the *inductive type* T generated by F in two ways:

- T with its constructor $c : FT \to T$ form an initial object in the category F-alg of F-algebras
- T satisfies an induction principle for F

Initiality

Define category F-alg:

- objects:

$$|F\text{-alg}| : \text{Type}$$
$$|F\text{-alg}| :\equiv (X : \text{Type}) \times (\theta : FX \to X)$$

- morphisms:

$$F\text{-alg}(\_,\_) : |F\text{-alg}| \to |F\text{-alg}| \to \text{Type}$$
$$F\text{-alg}((X,\theta),(Y,\rho)) :\equiv (f : X \to Y) \times (f_0 : f \circ \theta = \rho \circ Ff)$$

Note that the computation rule holds up to propositional equality.

Definition

An object X of category $\mathcal{C}$ is *(homotopy) initial* if we have:

$$(Y : |\mathcal{C}|) \to \text{is-contr } \mathcal{C}(X, Y)$$

Inductive type T is the carrier of the initial object of F-alg with its constructor $c : FT \to T$ as its algebra structure.

Induction principle

T : Type with constructor $c : FT \to T$ satisfies the induction principle if for any *algebra family*:

- $P : T \to \text{Type}$
- $m : (x : FT) \times \Box_F\ P\ x \to P\ (c\ x)$

we get a *dependent algebra morphism*:

- $s : (x : T) \to P\ x$
- $s_0 : (x : FT) \to s\ (c\ x) = m\ x\ (\bar{F}\ s\ x)$

Notation

- $\Box_F\ P : FT \to \text{Type}$ lifts the family P on T to FT
- $\bar{F} : ((x : T) \to P\ x) \to (x : FT) \to \Box_F\ P\ x$ lifts dependent functions on P to $\Box_F\ P$

Initiality versus induction in type theory

- For ordinary inductive types initiality and induction have been shown to be equivalent (Sojakova et al., 2012)
- For initiality we only need objects and morphisms
- For the induction principle we need more structure...

Induction – categorically

Instead of proving:

$$\text{initiality} \iff \text{induction}$$

directly, we introduce an intermediate concept:

$$\text{initiality} \iff \textit{section}\text{ induction} \iff \text{induction}$$

Definition

An object $X : |\mathcal{C}|$ satisfies the *section induction principle* if for every $Y : |\mathcal{C}|$ any $p : \mathcal{C}(Y, X)$ has a section $s : \mathcal{C}(X, Y)$

Initiality implies induction – categorically

Suppose $X : |\mathcal{C}|$ initial, then for any Y with $p : \mathcal{C}(Y, X)$ we have a unique $s : \mathcal{C}(X, Y)$ which is a section of p:

$$\begin{tikzcd} Y \arrow[d, "p"] \\ X \end{tikzcd}$$

Initiality implies induction – categorically

Suppose $X : |\mathcal{C}|$ initial, then for any Y with $p : \mathcal{C}(Y, X)$ we have a unique $s : \mathcal{C}(X, Y)$ which is a section of p:

$$
\begin{tikzcd}
X \arrow[r, "s"] & Y \arrow[d, "p"] \\
 & X
\end{tikzcd}
$$

Initiality implies induction – categorically

Suppose $X : |\mathcal{C}|$ initial, then for any Y with $p : \mathcal{C}(Y, X)$ we have a unique $s : \mathcal{C}(X, Y)$ which is a section of p:

$$\begin{tikzcd} X \arrow[r, "s"] \arrow[dr, "\mathrm{Id}_X"'] & Y \arrow[d, "p"] \\ & X \end{tikzcd}$$

Induction implies initiality – categorically

Suppose we have $X : |\mathcal{C}|$ that satisfies the section induction principle. Assuming $\mathcal{C}$ has products, then for any Y we have the projection:

$$X \times Y \xrightarrow{\pi_1} X$$

which has a section s. Define $f : \mathcal{C}(X, Y)$ as the composite:

$$X \xrightarrow{s} X \times Y \xrightarrow{\pi_2} Y$$

We have to show that any other $g : \mathcal{C}(X, Y)$ is equal to this f. Taking the equaliser of f with any g, we get:

$$\begin{tikzcd} X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \end{tikzcd}$$

Induction implies initiality – categorically

Suppose we have $X : |\mathcal{C}|$ that satisfies the section induction principle. Assuming $\mathcal{C}$ has products, then for any Y we have the projection:

$$X \times Y \xrightarrow{\pi_1} X$$

which has a section s. Define $f : \mathcal{C}(X, Y)$ as the composite:

$$X \xrightarrow{s} X \times Y \xrightarrow{\pi_2} Y$$

We have to show that any other $g : \mathcal{C}(X, Y)$ is equal to this f. Taking the equaliser of f with any g, we get:

$$\begin{tikzcd} E \arrow[r, "e"] & X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \end{tikzcd}$$

Induction implies initiality – categorically

Suppose we have $X : |\mathcal{C}|$ that satisfies the section induction principle. Assuming $\mathcal{C}$ has products, then for any Y we have the projection:

$$X \times Y \xrightarrow{\pi_1} X$$

which has a section s. Define $f : \mathcal{C}(X, Y)$ as the composite:

$$X \xrightarrow{s} X \times Y \xrightarrow{\pi_2} Y$$

We have to show that any other $g : \mathcal{C}(X, Y)$ is equal to this f. Taking the equaliser of f with any g, we get:

$$\begin{tikzcd}
E \arrow[r, "e"] & X \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & Y \\
X \arrow[u, "s"] \arrow[ur, "\mathrm{Id}_X"'] & &
\end{tikzcd}$$

Category structure needed

To formalise the previous arguments, we need the following:

- For initiality: objects and morphisms
- For sections:
  - Composition
  - Identity morphisms
- To show that initiality and section principle coincide:
  - Associativity
  - Identity laws
  - Products
  - Equalisers

Category structure needed - ordinary inductive types

Given $F : \text{Type} \to \text{Type}$ as a *container*, we can define F-alg as:

- objects:

$$|F\text{-alg}| : \text{Type}$$
$$|F\text{-alg}| :\equiv (X : \text{Type}) \times (\theta : FX \to X)$$

- morphisms:

$$F\text{-alg}(\_,\_) : |F\text{-alg}| \to |F\text{-alg}| \to \text{Type}$$
$$F\text{-alg}((X, \theta), (Y, \rho)) :\equiv (f : X \to Y) \times (f_0 : f \circ \theta = \rho \circ Ff)$$

- With some effort, we can define all the categorical structure needed
- However, categorical laws are not satisfied strictly

Higher inductive types versus ordinary inductive types

Ordinary inductive type T with constructors

- $c_0 : F_0\,T \to T$
- $\vdots$
- $c_k : F_k\,T \to T$

where every F_i : Type $\to$ Type is a (strictly positive) functor.

Higher inductive types versus ordinary inductive types

Ordinary inductive type T with constructor:

- $c : F_0 T + \ldots + F_k T \to T$

where every F_i : Type $\to$ Type is a (strictly positive) functor.

Higher inductive types versus ordinary inductive types

Ordinary inductive type type T with constructor:

- $c : FT \to T$

where $F : \mathrm{Type} \to \mathrm{Type}$ is a (strictly positive) functor.

Higher inductive types versus ordinary inductive types

Higher inductive types, e.g. the circle S^1 has constructors:

- base : S^1
- loop : base $=_{S^1}$ base

Differences from an ordinary inductive type:

- Dependencies on previous constructors
- *Higher* constructors: target of constructors not always T, but can also be an iterated path space of T.

Single functor Type $\to$ Type no longer suffices

Specifying 1-HITs

A specification of a 1-HIT is either:

- empty (denoted ϵ)
- a specification s extended with a 0-constructor on alg_s,
- or a specification s extended with a 1-constructor on alg_s

Defined mutually with the type of specifications is the function that maps a specification to its category of algebras:

$$\mathsf{alg} : \mathsf{Spec} \to \mathsf{Cat}$$

For the empty specification $\mathsf{alg}_\epsilon :\equiv \mathsf{Type}$
We also define mutually the underlying functor U_s which gives us the underlying type of the algebra.

Specifying 1-HITs – category of algebras – 0-constructors

Given s : Spec, a 0-constructor is given by a functor:

$$F : \mathsf{alg}_s \to \mathsf{Type}$$

and the category $\mathsf{alg}_{(s,F)}$ is defined as having:

- objects:
  - $X : |\mathsf{alg}_s|$
  - $\theta : FX \to U_s X$
- morphisms $(X, \theta) \to (Y, \rho)$:
  - $f : \mathsf{alg}_s(X, Y)$
  - $f_0 : U_s f \circ \theta = \rho \circ Ff$

Specifying 1-HITs – category of algebras – 1-constructors

Given s : Spec, a 1-constructor is given by specifying its arguments:

$$F : \mathsf{alg}_s \to \mathsf{Type}$$

and the endpoints of the path, as natural transformations:

$$l, r : F \to U_s$$

The category of algebras is then:

- objects:
  - $X : |\mathsf{alg}_s|$
  - $\theta : l_X =_{U_s X} r_X$
- morphisms $(X, \theta) \to (Y, \rho)$:
  - $f : \mathsf{alg}_s(X, Y)$
  - f_0 :

$$\begin{tikzcd} Uf \circ l_X \arrow[r, dash, "Uf \circ \theta"] \arrow[d, dash, "\alpha_f"'] & Uf \circ r_X \arrow[d, dash, "\beta_f"] \\ l_Y \circ Ff \arrow[r, dash, "\rho \circ Ff"'] & r_Y \circ Ff \end{tikzcd}$$

Functors and natural transformations in type theory

- Functors $F : \mathsf{alg}_s \to \mathsf{Type}$ need to be *strictly positive*
- Strictly positive functors $\mathsf{Type} \to \mathsf{Type}$: *containers*
- For a given $\mathcal{C} : \mathit{Cat}$, strictly positive functors $\mathcal{C} \to \mathsf{Type}$: *$\mathcal{C}$-containers*
- Natural transformations between strictly positive functors: *container morphisms*
- Forgetful functors $U_s : \mathsf{alg}_s \to \mathsf{Type}$ are containers if they have a left adjoint

Containers on Type

Strictly positive functors Type $\to$ Type: containers

- Shapes S : Type
- Positions $T : S \to$ Type

$$
\begin{aligned}
&[\![S \lhd P]\!]_0 : \text{Type} \to \text{Type} \\
&[\![S \lhd P]\!]_0\ X :\equiv (s : S) \times (P\ s \to X)
\end{aligned}
$$

$$
\begin{aligned}
&[\![S \lhd P]\!]_1 : (X \to Y) \to [\![S \lhd P]\!]_0\ X \to [\![S \lhd P]\!]_0\ Y \\
&[\![S \lhd P]\!]_1\ f\ (s, t) :\equiv (s, f \circ t)
\end{aligned}
$$

The functor laws follow from the categorical laws of Type, i.e. they are satisfied *strictly*

$\mathcal{C}$-containers

Strictly positive functors $\mathcal{C} \to \mathrm{Type}$: $\mathcal{C}$-containers (or *familially representable*)

- Shapes $S : \mathrm{Type}$
- Positions $T : S \to |\mathcal{C}|$

with

$$\llbracket S \lhd P \rrbracket_0 : \mathcal{C} \to \mathrm{Type}$$
$$\llbracket S \lhd P \rrbracket_0 \; X :\equiv (s : S) \times \mathcal{C}(P\ s, X)$$

$$\llbracket S \lhd P \rrbracket_1 : \mathcal{C}(X, Y) \to \llbracket S \lhd P \rrbracket_0 \; X \to \llbracket S \lhd P \rrbracket_0 \; Y$$
$$\llbracket S \lhd P \rrbracket_1 \; f \; (s, t) :\equiv (s, f \circ t)$$

The functor laws follow from the categorical laws of $\mathcal{C}$, i.e. associativity and identity laws

$\mathcal{C}$-container morphisms

Natural transformations between containers: *container morphisms*:
For containers $S \lhd P$ and $T \lhd Q$, container morphisms are:

$$(f : S \to T) \times (g : (a : S) \to \mathcal{C}(Q\ (f\ a), P\ a))$$

with

$$\begin{aligned}
&\text{apply } (f, g) : (X : |\mathcal{C}|) \to [\![S \lhd P]\!]_0\ X \to [\![T \lhd Q]\!]_0\ X \\
&\text{apply } (f, g)\ X\ (s, t) :\equiv (f\ s, t \circ (g\ s))
\end{aligned}$$

Naturality follows from associativity of $\mathcal{C}$

0-HITs and coherence

Consider a specification with three 0-constructors, i.e. the category of algebras has objects:

- X : Type
- $\theta_0 : F_0 X \to X$
- $\theta_1 : F_1(X, \theta_0) \to X$
- $\theta_2 : F_2(X, \theta_0, \theta_1) \to X$

Supposing all functors F_i are given as containers, then:

$$
\begin{aligned}
& \text{identity morphisms in } F_2\text{-alg} \\
\Leftarrow\ & \text{identity laws of functor } F_2 \\
\Leftarrow\ & \text{identity laws of category } F_1\text{-alg} \\
\Leftarrow\ & \text{coherence of identity laws of } F_0\text{-alg and composition law } F_1 \\
\Leftarrow\ & \text{coherence of identity and associativity laws of } F_0\text{-alg}
\end{aligned}
$$

Coherence issues increase with the amount of constructors, even if we only have 0-constructors

1-HITs

We will look at 1-HITs T with constructors:

- $c_0 : F_0 T \to T$
- $c_1 : (x : F_1 T) \to c_0^* \ (l_T \ x) =_T c_0^* \ (r_T \ x)$

where:

- $F_0, F_1 : \text{Type} \to \text{Type}$ functors given as containers
- $F_0^* : \text{Type} \to \text{Type}$ is the free monad of F_0, also given as a container
- $c_0^* : F_0^* X \to X$ is the algebra c_0 lifted to the free monad F_0^*
- $l, r : F_1 \to F_0^*$ natural transformations given as container morphisms

1-HITs

We will look at 1-HITs T with constructors:

- $c_0 : F_0 T \to T$
- $c_1 : c_0^* \circ l_T =_{F_1 T \to T} c_0^* \circ r_T$

where:

- $F_0, F_1 : \text{Type} \to \text{Type}$ functors given as containers
- $F_0^* : \text{Type} \to \text{Type}$ is the free monad of F_0, also given as a container
- $c_0^* : F_0^* X \to X$ is the algebra c_0 lifted to the free monad F_0^*
- $l, r : F_1 \to F_0^*$ natural transformations given as container morphisms

1-HITs – algebras

Given a specification (F_0, F_1, l, r), the category of algebras has:

- objects:
  - $X : Type$
  - $\theta_0 : F_0 X \to X$
  - $\theta_1 : \theta_0^* \circ l_X = \theta_0^* \circ r_X$
- morphisms $(X, \theta_0, \theta_1) \to (Y, \rho_0, \rho_1)$:
  - $f : X \to Y$
  - $f_0 : f \circ \theta_0 = \rho_0 \circ F_0 f$
  - $f_1 :$

$$
\begin{tikzcd}
f \circ \theta_0^* \circ l_X \arrow[r, dash, "f \circ \theta_1"] \arrow[d, dash, "f_0^* \circ l_X"'] & f \circ \theta_0^* \circ r_X \arrow[d, dash, "f_0^* \circ r_X"] \\
\rho_0^* \circ l_Y \circ F_1 f \arrow[r, dash, "\rho_1 \circ F_1 f"'] & \rho_0^* \circ r_Y \circ F_1 f
\end{tikzcd}
$$

1-HITs – algebras

Defining the category structure is involved:

- F_0, F_1 and l, r satisfy their respective laws strictly,
- $\_^* : |F\text{-alg}| \to |F^*\text{-alg}|$ is not strictly functorial however
- We need to show that its functoriality is coherent with the category structure
- There is a lot of path algebra involved
- Cubical methods from the `HoTT-Agda` library make life a bit easier

1-HITs – induction principle

We need to define algebra families and dependent algebra morphisms
We have:

$$(X \to \mathsf{Type}) = (Y : \mathsf{Type}) \times (p : Y \to X)$$

as witnessed by:

$$\mathit{to}\ P :\equiv (\Sigma X P, \pi_1)$$
$$\mathit{from}\ (Y, p) :\equiv p^{-1}$$

Under this equivalence, *sections* correspond to *dependent functions*

We can derive algebra families and dependent algebra families by finding similar equivalences, replacing types and functions with algebras and algebra morphisms.

1-HITs – induction principle – families

Suppose $(X, \theta_0, \theta_1) : |\mathrm{alg}_{(F_0, F_1, l, r)}|$, then the type of algebra families over (X, θ_0, θ_1) is an $M : (X \to \mathrm{Type}) \to \mathrm{Type}$ that satisfies satisfies the equation:

$$
\begin{aligned}
(P : X \to \mathrm{Type}) \times M\ P = {} & (Y : \mathrm{Type}) \\
& \times (\rho : FY \to Y) \\
& \times (p : Y \to X) \\
& \times (p_0 : \ldots) \\
& \times (p_1 : \ldots)
\end{aligned}
$$

1-HITs – induction principle – families

We can solve the previous equation by applying the witnesses of the equivalence $(P : X \to \text{Type}) = (Y : \text{Type}) \times (p : Y \to X)$. An algebra family then consists of:

$$
\begin{aligned}
& (P : X \to \mathit{Type}) \\
\times\ & (m_0 : (x : F_0 X) \times \Box_{F_0}\ P\ x \to P\ (\theta_0\ x)) \\
\times\ & (m_1 : (x : F_1 X) \times (y : \Box_{F_1}\ P\ x) \\
& \quad \to m_0^*\ (l^d(x, y)) = m_0^*\ (r^d(x, y))\ [P \downarrow \theta_1\ x])
\end{aligned}
$$

where

- $\Box_F$ satisfies $F(\Sigma X P) = \Sigma(FX)(\Box_F\ P)$
- $m_0^* : (x : F_0^* X) \times \Box_{F_0^*}\ P\ x \to P\ (\theta_0^*\ x)$
- $l^d, r^d : (x : F_1 X) \times \Box_{F_1}\ P\ x \to (x : F_0^* X) \times \Box_{F_0^*}\ P\ x$

1-HITs – induction principle – dependent morphisms

We can also figure out what under this equivalence the sections amount to and we arrive at the following, given an algebra family:

- $P : X \to Type$
- $m_0 : (x : F_0 X) \times \Box_{F_0}\ P\ x \to P\ (\theta_0\ x)$
- $m_1 : (x : F_1 X) \times (y : \Box_{F_1}\ P\ x)$
  $\to m_0^*\ (l^d(x, y)) = m_0^*\ (r^d(x, y))\ [P \downarrow \theta_1\ x]$

a dependent algebra morphism over (P, m_0, m_1) consists of:

- $s : (x : X) \to P\ x$
- $s_0 : (x : F_0 X) \to s\ (\theta_0\ x) = m_0\ x\ (\bar{F}_0\ s\ x)$
- $s_1 : (x : F_1 X) \to$

$$\begin{tikzcd}
s\ (\theta_0^*\ (l\ x)) \arrow[r, "s\ (\theta_1\ x)", "{[P \downarrow \theta_1\ x]}"'] \arrow[d, "s_0^*\ (l\ x)"'] & s\ (\theta_0^*\ (r\ x)) \arrow[d, "s_0^*\ (r\ x)"] \\
m_0^*(l^d(x, \bar{F}_1\ s\ x)) \arrow[r, "m_1\ x\ (\bar{F}_1\ s\ x)", "{[P \downarrow \theta_1\ x]}"'] & m_0^*(r^d(x, \bar{F}_1\ s\ x))
\end{tikzcd}$$

Conclusions

- We can
  - define 1-HITs in type theory
  - define induction for a class of them
  - show that for this class induction and initiality coincide
- Coherence issues increase with the number of constructors
- Dealing with these generally requires heavy machinery, e.g. $(\infty, 1)$-categories
- Agda formalisation is a work in progress: `https://github.com/gdijkstra/homotopy-initiality`