The Construction of Set-Truncated Higher Inductive Types

Niels van der Weide Herman Geuvers

Radboud University

June 7, 2019

Homotopy Type Theory: Another Perspective on Types

Its features are:

- Equality is proof relevant (*goodbye UIP*)

Homotopy Type Theory: Another Perspective on Types

Its features are:

- Equality is proof relevant (*goodbye UIP*)
- Interpretation in simplicial sets (*types are spaces, terms are points, equalities are paths*)

Homotopy Type Theory: Another Perspective on Types

Its features are:

- Equality is proof relevant (*goodbye UIP*)
- Interpretation in simplicial sets (*types are spaces, terms are points, equalities are paths*)
- Univalence Axiom (*equality of types is isomorphism*)

Homotopy Type Theory: Another Perspective on Types

Its features are:

- Equality is proof relevant (*goodbye UIP*)
- Interpretation in simplicial sets (*types are spaces, terms are points, equalities are paths*)
- Univalence Axiom (*equality of types is isomorphism*)
- **Higher Inductive Types** (*building spaces*)

Higher Inductive Types? What are they?

Inductive Types: a type generated by constructors for its points
For example, $\mathbb{N}$ is generated by $Z : \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$.

Higher Inductive Types? What are they?

Higher Inductive Types (HIT): a type generated by constructors for its points **and paths** (equalities)
For example, X is generated by $Z : X$ and $S : X \to X$ **and**

$$m : \prod_{n:X} n = S(S\ n).$$

Higher Inductive Types? What are they?

Higher Inductive Types (HIT): a type generated by constructors for its points **and paths** (equalities)
For example, X is generated by $Z : X$ and $S : X \to X$ **and**

$$m : \prod_{n:X} n = S(S\ n).$$

Now we have two ways to prove $S\ Z = S^3\ Z$:

- use $m(S\ Z)$
- use $m(Z)$ and S preserves equality

Higher Inductive Types? What are they?

Higher Inductive Types (HIT): a type generated by constructors for its points, paths (equalities), **and homotopies (equalities of equalities)**

For example, $\mathbb{N}_2$ is generated by $Z : \mathbb{N}_2$ and $S : \mathbb{N}_2 \to \mathbb{N}_2$ **and two equalities:**

$$m : \prod_{n:\mathbb{N}_2} n = S(S\ n),$$

$$t : \prod_{n,m:\mathbb{N}_2} \prod_{p,q:n=m} p = q.$$

But let's not get too high

In this talk, we only consider HITs

- constructed by giving points and equalities
- and with the constructor

$$t : \prod_{x,y:X} \prod_{p,q:x=y} p = q.$$

We call these **set-truncated HITs**.
Recall: a set is a type for which all $p, q : x = y$ are the same.

Now, what did we do?

Our result: all set-truncated HITs exists if two simple ones exist:

- The **quotient**. Given A and an equivalence relation R on A, identify the points in A via R.
- Given a type A, then the **propositional truncation** $||A||$ is A with all its points identified.

Note: we construct them in HoTT. Formalization in UniMath.

A Bird's-Eye View of our Construction

The steps:

- Define signatures of HITs (*how to describe them*)
- Give categories of algebras in sets **and setoids** (*the introduction rules*)
- Use initial algebra semantics (*showing induction by initiality*)
- Relate those categories by an adjunction (*the main idea*)
- Construct the initial algebra in setoids (*where we get to work*)

Recall:

- a set is a type for which all proofs of equality are equal
- a setoid is a set with an equivalence relation

Wells of Inspiration

Note

- Initial algebra semantics is used: QIITs (by Altenkirch, Capriotti, Dijkstra, Kraus, Forsberg) and W-suspensions (Sojakova).
- To obtain the adjunction, we use a result by Hermida and Jacobs.
- The construction of the initial algebra in setoids is an adaption of work by Dybjer and Moenclaey.

Let's Get Started: Signatures for HITs

Recall: $\mathbb{N}_2$ is generated by $Z : \mathbb{N}_2$ and $S : \mathbb{N}_2 \to \mathbb{N}_2$ and two equalities:

$$m : \prod_{n:\mathbb{N}_2} n = S(S\ n),$$

$$t : \prod_{n,m:\mathbb{N}_2} \prod_{p,q:n=m} p = q.$$

To define signatures of HITs, note the following:

- We must describe arguments of the point constructors
- We must describe the possible endpoints for the equalities, for which we can refer to the point constructor

Then the introduction/elimination rules are derived.

First Ingredient: Point Constructors

Same idea as for inductive types.
We use **finitary polynomials**. These are generated by

- The *identity*
- Given a set X, we have a *constant* polynomial on X
- Given two polynomials, we have their *product* and *sum*

Recall $\mathbb{N}$ generated by $Z : \mathbb{N}$ and $S : \mathbb{N} \to \mathbb{N}$.
This is represented by $F(X) = 1 + X$

Second Ingredient: Path Constructors

Recall path of $\mathbb{N}_2$:

$$m : \prod_{n:\mathbb{N}_2} n = S(S\ n),$$

Second Ingredient: Path Constructors

Recall path of $\mathbb{N}_2$:

$$m : \prod_{n:\mathbb{N}_2} n = S(S\ n),$$

There are two ingredients for path constructors:

- A polynomial representing the *source* of the equation
- Two endpoints representing the *sides* of the equation

The endpoints represent all possible left- and right hand sides of equations. Note: they *depend on the point constructor and source*.

Second Ingredient: Path Constructors

Recall path of $\mathbb{N}_2$:

$$m : \prod_{n:\mathbb{N}_2} n = S(S\ n),$$

There are two ingredients for path constructors:

- A polynomial representing the *source* of the equation
- Two endpoints representing the *sides* of the equation

The endpoints represent all possible left- and right hand sides of equations. Note: they *depend on the point constructor and source*.
Main challenge: define tthe type of endpoints

- it's a type $\mathcal{E}_P(S)$ depending polynomials P, S
- inductively generated by 10 constructors

Putting it together: Signatures

Definition

A **HIT signature** consists of

- A polynomial P representing its *point constructor*
- A type J representing *labels for its path constructors*
- For each $j : J$, a polynomial S_j representing the *arguments of the path constructors*
- For each $j : J$, two endpoint with source S_j using constructors from P representing the *path*

Note: we will interpret signatures in sets, so we always have a constructor

$$t : \prod_{n,m:X} \prod_{p,q:n=m} p = q.$$

How does this give rise to a notion of HITs?

- All we did so far, was say how to describe the constructors.
- Soon we discuss algebras, which describe the introduction rules.
- You also need to formulate an induction principle.
- Details on that are in the paper/formalization.

Intermezzo: Some Examples

- Signature $\mathbb{N}_2$ (discussed before)
- Similarly, we can define a signature for the integers.
- We can define signatures of groups, rings, and so on
- Also: polynomials, free algebras

A Bird's-Eye View of our Construction

The steps:

- Define signatures of HITs (how to describe them)
- **Give categories of algebras in sets and setoids (the introduction rules)**
- Use initial algebra semantics (showing induction by initiality)
- Relate those categories by an adjunction (the main idea)
- Construct the initial algebra in setoids (where we get to work)

Algebra in Two Steps: the **pre**algebras

Note: each polynomial P gives rise to a functor $\overline{P}$ on **Set**.
Now let S be a signature and let P be the point constructor.

- A **prealgebra** on a signature consists of a set X with a map $\overline{P}\,X \to X$.
- This forms a category $\mathbf{PreAlgSet}(P)$ whose morphisms are

$$\begin{tikzcd} \overline{P}\,X \arrow[r, "P\,f"] \arrow[d] & \overline{P}\,Y \arrow[d] \\ X \arrow[r, "f"'] & Y \end{tikzcd}$$

- Note: we have a functor $U : \mathbf{PreAlgSet}(P) \to \mathbf{Set}$

Algebra in Two Steps: the **actual** algebras

Now come the equations.
Let e be an endpoint with source S, and constructor P. Note we have functors:

$$\mathbf{PreAlgSet}(P) \xrightarrow{U} \mathbf{Set} \xrightarrow{S} \mathbf{Set}$$

Then e gives a natural transformation from $S \circ U$ to U.
Algebras are defined as a full subcategory of prealgebras.

We can repeat this story for setoids

It will be much the same but with setoids instead.

A Bird's-Eye View of our Construction

The steps:

- Define signatures of HITs (how to describe them)
- Give categories of algebras in sets and setoids (the introduction rules)
- **Use initial algebra semantics (showing induction by initiality)**
- Relate those categories by an adjunction (the main idea)
- Construct the initial algebra in setoids (where we get to work)

The Induction Principle and Initial Algebras

- HITs satisfy an *induction principle* (IP)
- The IP is formulated with *displayed algebras* (aka fibred algebras)
- To verify IP, we use *initial algebra semantics*
- It says: **induction follows from initiality**

A Bird's-Eye View of our Construction

The steps:

- Define signatures of HITs (how to describe them)
- Give categories of algebras in sets and setoids (the introduction rules)
- Use initial algebra semantics (showing induction by initiality)
- **Relate those categories by an adjunction (the main idea)**
- Construct the initial algebra in setoids (where we get to work)

The moment we waited for: Adjunctions

Let's start with the "base". Write

- **Set** and **Setoid** for the categories of sets and setoids respectively
- **Quot** sends a setoid (X, R) to the quotient of X by R
- **Path** sends a set X to the setoid $(X, =)$

Then we have an adjunction

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "\mathbf{Path}"] \arrow[r, phantom, "\top"] & \mathbf{Setoid} \arrow[l, bend left, "\mathbf{Quot}"] \end{tikzcd}$$

The moment we waited for: Adjunctions

Lift it to an adjunction on prealgebras (Hermida and Jacobs)

$$\begin{tikzcd}
\mathbf{PreAlgSet} \arrow[rr, bend left, "\mathbf{PreAlgPath}"] \arrow[rr, phantom, "\top"] \arrow[d] & & \mathbf{PreAlgSetoid} \arrow[ll, bend left, "\mathbf{PreAlgQuot}"] \arrow[d] \\
\mathbf{Set} \arrow[rr, bend left, "\mathbf{Path}"] \arrow[rr, phantom, "\top"] & & \mathbf{Setoid} \arrow[ll, bend left, "\mathbf{Quot}"]
\end{tikzcd}$$

Here we need the polynomials are finitary.
For infinitary case, we need AC (Chapman, Uustalu, Veltri)

The moment we waited for: Adjunctions

Next we lift it to the level of algebras

$$
\begin{tikzcd}
\mathbf{AlgSet} \arrow[rr, bend left, "\mathbf{AlgPath}"] \arrow[rr, phantom, "\top"] \arrow[d] & & \mathbf{AlgSetoid} \arrow[ll, bend left, "\mathbf{AlgQuot}"] \arrow[d] \\
\mathbf{PreAlgSet} \arrow[rr, bend left, "\mathbf{PreAlgPath}"] \arrow[rr, phantom, "\top"] \arrow[d] & & \mathbf{PreAlgSetoid} \arrow[ll, bend left, "\mathbf{PreAlgQuot}"] \arrow[d] \\
\mathbf{Set} \arrow[rr, bend left, "\mathbf{Path}"] \arrow[rr, phantom, "\top"] & & \mathbf{Setoid} \arrow[ll, bend left, "\mathbf{Quot}"]
\end{tikzcd}
$$

Before we continue, where are we?

- Define signatures of HITs (how to describe them)
- Give categories of algebras in sets and setoids (the introduction rules)
- Use initial algebra semantics (showing induction by initiality)
- Relate those categories by an adjunction (the main idea)
- **Construct the initial algebra in setoids (where we get to work)**

Recall that our goal was to construct HITs. However, so far, no constructing actually happened.
We only did preparatory work, which shows that

We obtain HITs by constructing initial algebras in setoids!

Finishing the proof: The Initial Algebra in Setoids

- Dybjer and Moenclaey give an interpretation of HITs in the setoid and groupoid model
- The set is inductively generated by the point constructor
- The equivalence relation is the inductive family generated by the path constructors

Here we need the propositional truncation.

- In HoTT, equivalence relations take values in propositions
- Recall: a proposition is a type for which all inhabitants are equal
- To force the generated relation to be a proposition, we truncate it

Time to Wrap Up

Theorem

In UF with quotients and the propositional truncation, we can construct all finitary set-truncated HITs.

We obtain two consequences of our construction:

- A uniqueness principle for HITs (*go univalence*)
- A characterization of the path space of HITs

Where to go from here?

Remove the truncatedness restriction.

- First step: go to 1-types/groupoids
- Their structure is encapsulated by **bicategories**
- Imitate this construction bicategorically

Formalization: `https://github.com/nmvdw/SetHITs`.

References

1. T. Altenkirch, P. Capriotti, G. Dijkstra, N. Kraus, and F.N. Forsberg. Quotient Inductive-Inductive Types. In FOSSACS 2018, pages 293–310.
2. J. Chapman, T. Uustalu, and N. Veltri. Quotienting the Delay Monad by Weak Bisimilarity. MSCS, 29(1):67–92, 2019.
3. P. Dybjer and H. Moeneclaey. Finitary Higher Inductive Types in the Groupoid Model. ENTCS, 336:119–134.
4. C. Hermida and B. Jacobs. Structural Induction and Coinduction in a Fibrational Setting. *Information and computation*, 145(2):107–152.
5. K. Sojakova. Higher Inductive Types as Homotopy-Initial Algebras. In POPL 2015, pages 31–42.
6. V. Voevodsky, B. Ahrens, D. Grayson, et al. UniMath — a computer-checked library of univalent mathematics. available at `https://github.com/UniMath/UniMath`.