Issue IV: Higher Inductive Types

Maxim Sokhatsky [1]

[1] National Technical University of Ukraine
Igor Sikorsky Kyiv Polytechnical Institute

April 22, 2025

Abstract

CW-complexes are key in homotopy type theory (HoTT) and are encoded in cubical type checkers as higher inductive types (HITs). Like recursive trees for (co)inductive types, HITs represent CW-complexes. An HIT defines a CW-complex using cubical composition as an initial algebra element in a cubical model. We explore HIT motivation, their topological role, and implementation in Agda Cubical, focusing on infinity constructors.

Keywords: Cellular Topology, Cubical Type Theory, HITs

Contents

1 Higher Inductive Types

CW-complexes are central in HoTT and appear in cubical type checkers as HITs. Unlike inductive types (recursive trees), HITs encode CW-complexes, capturing points (0-cells) and higher paths (n-cells). Defining an HIT specifies a CW-complex via cubical composition, an initial algebra in a cubical model.

Definition 1. (Pushout). Pushouts are key HITs for cell attachments and cofibrant objects.

```
data pushout (A B C: U) (f: C -> A) (g: C -> B)
   = po1 (_: A)
   | po2 (_: B)
   | po3 (c: C) <i> [ (i = 0) -> po1 (f c) ,
                      (i = 1) -> po2 (g c) ]
```

Definition 2. (Spheres and Disks). Spheres are HITs with higher-dimensional paths.

```
data S1
   = base
   | loop <i> [ (i = 0) -> base,
                (i = 1) -> base ]

data S2
   = point
   | surf <i j> [ (i = 0) -> point, (i = 1) -> point,
                  (j = 0) -> point, (j = 1) -> point ]
```

2 CW-Complexes

CW-complexes are spaces built by attaching cells of increasing dimension. In HoTT, they are encoded as HITs, with cells as constructors for points and paths.

Definition 3. (Cell Attachment). Attaching an n-cell to a space X along $f : S^{n-1} \to X$ is a pushout:

$$
\begin{tikzcd}
S^{n-1} \arrow[r, "f"] \arrow[d, "\iota"] & X \arrow[d, "j"] \\
D^n \arrow[r, "g"] & X \cup_f D^n
\end{tikzcd}
$$

Here, $\iota : S^{n-1} \hookrightarrow D^n$ is the boundary inclusion, and $X \cup_f D^n$ is the pushout, gluing an n-cell to X via f. The result depends on the homotopy class of f.

Definition 4. (CW-Complex). A CW-complex is a space X built inductively by attaching cells, with a skeletal filtration:

- The (-1)-skeleton is $X_{-1} = \varnothing$.

- For $n \geq 0$, the n-skeleton X_n is obtained by attaching n-cells to X_{n-1}. For indices J_n and maps $\{f_j : S^{n-1} \to X_{n-1}\}_{j \in J_n}$, X_n is the pushout:

$$\begin{tikzcd} \coprod_{j \in J_n} S^{n-1} \arrow[r, "\coprod f_j"] \arrow[d, "\coprod \iota_j"] & X_{n-1} \arrow[d, "i_n"] \\ \coprod_{j \in J_n} D^n \arrow[r, "\coprod g_j"] & X_n \end{tikzcd}$$

  where $\coprod_{j \in J_n} S^{n-1}$, $\coprod_{j \in J_n} D^n$ are disjoint unions, and $i_n : X_{n-1} \hookrightarrow X_n$ is the inclusion.

- X is the colimit of:

$$\varnothing = X_{-1} \hookrightarrow X_0 \hookrightarrow X_1 \hookrightarrow \ldots \hookrightarrow X,$$

  with X_n the n-skeleton, and $X = \mathrm{colim}_{n \to \infty} X_n$. The sequence is the skeletal filtration.

In HoTT, CW-complexes are HITs, with constructors for cells and path constructors for gluing.

Example 1. (Sphere as a CW-Complex). The n-sphere S^n is a CW-complex with one 0-cell and one n-cell:

- $X_0 = \{\text{base}\}$, a point.
- $X_k = X_0$ for $0 < k < n$, no cells added.
- X_n: Attach an n-cell to $X_{n-1} = \{\text{base}\}$ along $f : S^{n-1} \to \{\text{base}\}$:

```
data Sn (n: Nat)
  = base
  | cell <i1 ... in> [ (i1 = 0) -> base, (i1 = 1) -> base,
                       ...,
                       (in = 0) -> base, (in = 1) -> base ]
```

  The cell constructor glues boundaries to base, yielding S^n.

3 Motivation for Higher Inductive Types

HITs in HoTT enable direct encoding of topological spaces like CW-complexes. In homotopy theory, spaces are built by gluing cells via attaching maps. HoTT views types as spaces, elements as points, and equalities as paths, making HITs a natural fit. Standard inductive types cannot capture higher homotopies, but HITs allow constructors for points and paths.

For example, the circle S^1 (Definition 2) has a base point and a loop, encoding its fundamental group $\mathbb{Z}$. HITs avoid set-level quotients, preserving HoTT's synthetic nature. In cubical type theory, paths are intervals (e.g., $< i >$) with computational content, unlike propositional equalities, enabling efficient type checking in tools like Agda Cubical.

3.1 Key Entities

HITs encode complex spaces via cell-like constructors. Below are key examples:

3.1.1 Sphere

The n-sphere S^n is a CW-complex with one 0-cell and one n-cell, as in the CW-complex example. In HIT form, for $n = 1$:

```
data S1
  = base
  | loop <i> [ (i = 0) -> base, (i = 1) -> base ]
```

This captures the loop structure of S^1, with higher spheres (e.g., S^2) adding multi-dimensional paths.

3.1.2 Torus

The torus T^2 is a 2D CW-complex with one 0-cell, two 1-cells, and one 2-cell:

```
data Torus
  = point
  | loop1 <i> [ (i = 0) -> point, (i = 1) -> point ]
  | loop2 <i> [ (i = 0) -> point, (i = 1) -> point ]
  | square <i j> [ (i = 0) -> loop1 @ j, (i = 1) -> loop1 @ j,
                   (j = 0) -> loop2 @ i, (j = 1) -> loop2 @ i ]
```

The square ensures loops commute, encoding $\pi_1(T^2) = \mathbb{Z} \times \mathbb{Z}$.

3.1.3 Suspension

The suspension ΣA of a type A adds two points and paths from A to them:

```
data Suspension (A: U)
  = north
  | south
  | merid (a: A) <i> [ (i = 0) -> north, (i = 1) -> south ]
```

For $A = S^0$, this yields S^1, showing how HITs build higher spheres.

3.2 Other Examples

Additional HITs demonstrate versatility:

- **Truncation**: The n-truncation $\|A\|_n$ collapses homotopies above dimension n:

  ```
  data Trunc (n: Nat) (A: U)
    = hub (a: A)
    | spoke (p: Path A a b) <i> [ (i = 0) -> hub a, (i = 1) -> hub b ]
  ```

  This ensures connectivity up to dimension n.

4 HITs with Infinity Constructors

Some HITs need infinite constructors for spaces like Eilenberg-MacLane spaces or the infinite sphere S^∞:

```
data SInf
  = base
  | loopn (n: Nat) <i> [ (i = 0) -> base, (i = 1) -> base ]
```

Here, loopn adds a path per natural number, forming an infinite 1-cell family. Challenges include:

1. **Type Checking**: Infinite constructors complicate termination and coverage. 2. **Computation**: Cubical type theory must ensure compositions are computable. 3. **Expressivity**: Needed for higher homotopy groups, but risks non-constructivity.

4.1 Agda Cubical and Infinity Constructors

Agda Cubical uses cubical primitives to handle HITs:

- **Cubical Primitives**: Intervals $I = [0, 1]$, PathP, and operations (hcomp, glue) give computational content. For S^1:

```
data S1 : Type where
  base : S1
  loop : PathP (\lambda i -> S1) base base
```

This extends to infinite constructors via indices.

- **HIT Support**: Indexed HITs emulate infinite families:

```
data InfHIT (A: Type) : Type where
  point : InfHIT A
  pathn : (n: Nat) -> PathP (\lambda i -> InfHIT A) point point
```

- **Normalization**: Reduces compositions to canonical forms using cubical coherence.

- **Limitations**: Infinite constructors slow type checking; Agda Cubical optimizes with strict evaluation.

For $K(\mathbb{Z}, 1)$, finite approximations manage complexity.

5 Conclusion

HITs encode CW-complexes in HoTT, bridging topology and type theory. They capture cell attachments, as seen in the CW-complex definition, with examples like spheres and tori. Infinity constructors extend HITs to infinite spaces, handled by Agda Cubical's primitives and indexed HITs. As cubical type theory grows, HITs will advance synthetic homotopy theory in proof assistants.

References

[1] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*, IAS, 2013.

[2] C. Cohen, T. Coquand, S. Huber, A. Mörtberg, *Cubical Type Theory*, Journal of Automated Reasoning, 2018.

[3] A. Mörtberg et al., *Agda Cubical Library*, `https://github.com/agda/cubical`, 2023.

[4] M. Shulman, *Higher Inductive Types in HoTT*, `https://arxiv.org/abs/1705.07088`, 2017.