Higher Inductive Types without Recursive Higher Constructors

MGS Christmas Seminar
Birmingham

Nicolai Kraus

University of Nottingham

17/12/15
Inductive types in Martin-Löf type theory

natural numbers

\( \mathbb{N} \) is a type with constructors

- \( z : \mathbb{N} \)
- \( S : \mathbb{N} \rightarrow \mathbb{N} \)

propositional equality

\( x = y \) is a type (for \( x, y : A \)).

Constructor:

\( \text{refl} : \forall x. x = x \)

\( \mathbb{N} \)-induction

for a family \( P : \mathbb{N} \rightarrow \mathcal{U} \),

\[
\begin{align*}
P(z) \\
\forall n. P(n) & \rightarrow P(Sn) \\
\end{align*}
\]

\( \text{ind} \Rightarrow \forall n. P(n) \)

\( =\)-induction

for \( P : (\Sigma_{x,y:A} x = y) \rightarrow \mathcal{U} \),

\[
\begin{align*}
\forall x. P(x, x, \text{refl}_x) \\
\text{ind} \Rightarrow \forall xyq. P(x, y, q) \\
\end{align*}
\]
Propositional equality ("="), examples

<table>
<thead>
<tr>
<th>symmetry</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sym</strong>: ( \forall xy. x = y \rightarrow y = x )</td>
</tr>
<tr>
<td>Construction: sym((x, x, \text{refl}_x)) :(\equiv) refl(_x).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>transitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>trans</strong>: ( \forall xyz. x = y \rightarrow y = z \rightarrow x = z )</td>
</tr>
<tr>
<td>Construction: trans((x, x, x, \text{refl}_x, \text{refl}_x)) :(\equiv) refl(_x).</td>
</tr>
</tbody>
</table>

“What’s the point? Everything is refl anyway...”
⇒ Try to prove *uniqueness of identity proofs* (UIP), i.e.
\[ \forall xy. \forall (p, q : x = y). p = q \]

“Sure, assume \((x, y, p)\) is just \((x, x, \text{refl}_x)\), then what we need is \( \forall x. \forall (q : x = x). \text{refl}_x = q \) and then...oh, we are stuck.”

Fact: **UIP is not derivable** (Hofmann-Streicher 1998).
Why not add UIP as an axiom?
Arguments for homotopy type theory (“alternative” to UIP):
○ want: if $A \simeq B$, then $A = B$ (without forgetting how)
e.g.: we want to substitute different representations of $\mathbb{N}$ for each other!
⇒ univalence (more abstract, more convenient to use)
○ “types are [behaved like] spaces”: transport intuition and results between homotopy theory and type theory;
allows other constructions, including synthetic homotopy theory
○ equalities are paths, and all type-theoretic statements are up to homotopy and continuous transformations –
everything automatically “non-evil”, it’s beautiful!
○ by the way:
  homotopy (type theory)
  (homotopy type) theory
Higher inductive types (HITs)

Constructors can construct elements, higher constructors can construct equalities. Example:

- **real numbers**

\[ \mathbb{R} \] is a type with constructors

\[ \text{rat} : \mathbb{Q} \to \mathbb{R} \]

\[ \text{lim} : (f : \mathbb{N} \to \mathbb{R}) \to \text{isCauchy}(f) \to \mathbb{R} \]

\[ \text{quot} : (u, v : \mathbb{R}) \to u \sim v \to u = v \]

isCauchy and \( \sim \) need to be defined at the same time (not shown here).

This is better behaved than a quotient – our \( \mathbb{R} \) is complete!
Higher inductive types (HITs)

$\mathbb{S}^1$ is given by the constructors

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

$\mathbb{S}^1$ behaves as one expects. E.g., its fundamental group is equivalent to $\mathbb{Z}$. The fundamental group is essentially $\text{base} =_{\mathbb{S}^1} \text{base}$. 
Higher inductive types (HITs)

**Propositional Truncation ("Squash", "bracket types")**

For a type $A$, the type $\|A\|$ is given by

$$\begin{align*}
\|A\| &= \{-\} : A \to \|A\| \\
h &= (x, y : \|A\|) \to x = \|A\| y
\end{align*}$$

- all elements of $\|A\|$ are equal
- $\|A\|$ is "the proposition that $A$ holds"
- can make "non-continuous" statement, e.g.
  $$\begin{align*}
  \prod_{x:S^1} x = \text{base} &\quad - \quad \text{contradiction} \\
  \prod_{x:S^1} \|x = \text{base}\| &\quad - \quad \text{provable}
  \end{align*}$$
# Recursive versus Non-Recursive HITs

## Propositional Truncation $\|A\|$:

| $|-| : A \to \|A\|$ |
| $h : (x, y : \|A\|) \to x =_{\|A\|} y$ |

**universal property $\|A\|$**

$$\|A\| \to B \quad \frac{}{A \to B}$$

*if $B$ is propositional*

## Pseudo-truncation $\langle A \rangle$:

| $\langle - \rangle : A \to \langle A \rangle$ |
| $t : (x, y : A) \to \langle x \rangle =_{\langle A \rangle} \langle y \rangle$ |

**universal property $\langle A \rangle$**

$$\langle A \rangle \to B \quad \frac{}{\Sigma (f : A \to B) . \text{wconst}(f)}$$

*for any $B$*

**note:** $\text{wconst}(f) \equiv \prod_{x, y : A} f a = f b$

$\langle A \rangle$ has several names:

- Altenkirch: “constant map classifier” (see u.p.)
- Coquand-Escardó: “generalised circle” ($\langle 1 \rangle \simeq S^1$)
- van Doorn: “one-step truncation” (later)
### Recursive versus Non-Recursive HITs

#### Propositional Truncation $\|A\|$  

$\|\_\| : A \to \|A\|$  

$h : (x, y : \|A\|) \to x =_{\|A\|} y$

**Universal property $\|A\|$**  

\[
\begin{align*}
\|A\| & \to B \\
\sum (f : A \to B) \cdot wconst(f) & \to B
\end{align*}
\]

if $B$ is propositional

#### Pseudo-truncation \(\langle A \rangle\)

\(\langle - \rangle : A \to \langle A \rangle\)

\(t : (x, y : A) \to x =_{\langle A \rangle} y\)

**Universal property \(\langle A \rangle\)**

\[
\begin{align*}
\langle A \rangle & \to B \\
\Sigma (f : A \to B) \cdot wconst(f) & \to B
\end{align*}
\]

for any $B$

**Note:** $wconst(f) :\equiv \prod_{x,y:A} f a = f b$

Recursion in path constructors makes elimination principles difficult to use! Do we actually need it?
Another HIT

Given an $\omega$-chain

$$A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} A_2 \xrightarrow{f_2} \ldots,$$

its sequential colimit $A_\omega$ is given by the constructors

$$\text{in} : (n : \mathbb{N}) \to A_n \to A_\omega$$

$$\text{glue} : (n : \mathbb{N}) \to (a : A_n) \to \text{in}_n(a) =_{A_\omega} \text{in}_{n+1}(f_n a)$$

This is a non-recursive HIT.
Higher inductive types (HITs)

Theorem (van Doorn)

The sequential colimit of

\[ A \xrightarrow{\langle - \rangle} \langle A \rangle \xrightarrow{\langle - \rangle} \langle \langle A \rangle \rangle \xrightarrow{\langle - \rangle} \langle \langle \langle A \rangle \rangle \rangle \xrightarrow{\langle - \rangle} \ldots \]

is propositional (and has the elimination principle of \( \| A \| \)).

Note: \( \langle \ldots \langle A \rangle \ldots \rangle \) is very complicated (homotopically).

Theorem (generalisation)

Given a chain \( A_0 \xrightarrow{f_0} A_1 \xrightarrow{f_1} A_2 \xrightarrow{f_2} \ldots \). If every \( f_i \) is weakly constant, then \( A_\omega \) is propositional (i.e. all its elements are equal).
Higher truncations

○ For every \( n \geq -2 \), we can define the \( n \)-truncation \( \|A\|_n \) as a HIT which trivialises all levels above \( n \)
○ to be precise: \( \|A\| \) is the case \( n \equiv -1 \) (index omitted)
○ let’s exchange the recursive constructor for a non-recursive one: we get general “pseudo-truncations” \( \langle A \rangle_n \), one constructor is \( \langle - \rangle_n : A \to \langle A \rangle_n \)
○ \( \langle - \rangle_n \) is not weakly constant in general

Consider the chain

\[
A \xrightarrow{\langle - \rangle_{-1}} \langle A \rangle_{-1} \xrightarrow{\langle - \rangle_0} \langle \langle A \rangle_{-1} \rangle_0 \xrightarrow{\langle - \rangle_1} \langle \langle \langle A \rangle_{-1} \rangle_0 \rangle_1 \xrightarrow{\langle - \rangle_2} \ldots
\]

Theorem

Every \( \langle - \rangle_n \) in the above chain is weakly constant (for inhabited \( A \)); and the sequential colimit of this chain has all the properties of \( \|A\| \).
Chain of higher pseudo-truncations

- the constructed chain converges: after \( n \) steps, the first \( n \) levels are “correct” (i.e. “conditionally \((n-1)\)-connected”)
- implies finite elimination principles for truncated types
- our chain is more minimalistic than van Doorn’s – we get a strict natural transformation

\[
\begin{align*}
A &\xrightarrow{\langle-\rangle_{-1}} \langle A\rangle_{-1} \xrightarrow{\langle-\rangle_0} \langle\langle A\rangle\rangle_{0} \xrightarrow{\langle-\rangle_1} \ldots \\
\langle A\rangle &\xrightarrow{\langle-\rangle} \langle\langle A\rangle\rangle \xrightarrow{\langle-\rangle} \langle\langle\langle A\rangle\rangle\rangle \xrightarrow{\langle-\rangle} \ldots
\end{align*}
\]

- can derive the finite elimination principles for the chaotic van Doorn chain (any cocone of the second chain gives one of the first).
Conclusion

**Conjecture**

*Every HIT can be represented without recursive path-constructors.*

- The general case is expected to be far more difficult.
- The conjecture is currently not even a precise statement: what is “every HIT”? – but that’s another topic...

Thank you for your attention!