An Introduction to Homotopy Type Theory

Eric Finster

GETCO 2022 Paris
Part I

Type Theory
The Setup of Classical Mathematics

- The mathematical universe consists of abstract collections called sets.

- Logic (connectives, rules of inference, ...) exists prior to the definition of the theory of sets.

- Properties of sets are axiomatized using this logic.
• Law of Excluded Middle

Connectives determined by their truth tables.

<table>
<thead>
<tr>
<th>V</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>∧</th>
<th>T</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

• Proofs are "external" to the theory
The theory is proof-irrelevant.
Some Criticisms of Set Theory

- Non-sensical but well-formed assertions:
  \[ \text{Is } \pi \text{ ?} \]

- Properties of objects depend on implementation:
  \[ \mathbb{N} = \{ 0, 3, 2^0 3, \ldots \} \quad \mathbb{N}' = \{ 0, 3, 2^0 3, \ldots \} \]

- Reasonable disagreement about axioms:
  \[ \text{AC? CH?} \]

- Non-constructive by default
Type Theory

- The mathematical universe consists of Statements and their Proofs.

- Proofs are gathered into collections based on what they prove.

- We write this as: Type (statement)

  Term → x : A (proof)
The Brouwer–Heyting–Kolmogorov (BHK) Interpretation

- A proof of \( \text{AvB} \) is either a proof of \( A \) or a proof of \( B \)
- A proof of \( \text{AnB} \) is a pair of a proof of \( A \) and a proof of \( B \)
- A proof of \( A \Rightarrow B \) is a function which assigns to any proof of \( A \) a proof of \( B \).
- There is no proof of \( \bot \)
- A proof of \( \neg A \) is a proof of \( A \Rightarrow \bot \)

Logical connectives explained by evidence.
Type Theory as an Implementation of BHK

- We make this idea precise by providing explicit syntax for constructing statements and their proofs.

\[
\begin{align*}
\text{a:A} & \quad \text{b:B} \\
\frac{}{a,b : A \times B} \\
\text{a:A} & \quad \text{b:B} \\
\frac{}{\text{in} \text{l} a : A \cup B} & \quad \frac{}{\text{in} \text{r} b : A \cup B} \\
\text{x:A} & \quad \text{b:B} \\
\frac{}{\lambda x.b : A \Rightarrow B}
\end{align*}
\]
The Natural Numbers

A proof \( n : \mathbb{N} \) can be thought of as the proof of the statement:

"I know a natural number"

\[
\begin{array}{c}
\hline
O : \mathbb{N} \\
\hline
S \circ : \mathbb{N} \\
\hline
\end{array}
\]

Ex: \( O : \mathbb{N} \)

So: \( \mathbb{N} \)

SSD: \( \mathbb{N} \)
Dependent Types

. So far we have only seen simple types.
\[ \forall \wedge \Rightarrow \neg \text{ IN Bool} \]

. But if types are to be a rich enough language for mathematics, we must also allow them to mention terms.
\[ 4 \leq 7 \quad \forall n: \mathbb{Z}. \, n^2 \geq 0 \]

. We call these dependent types.
Example

\[
\begin{array}{c}
\text{n : \(\mathbb{N}\)} \\
\text{m : \(\mathbb{N}\)} \\
\hline
\text{n \leq m : Type}
\end{array}
\]

Formation Rule

\[
\begin{array}{c}
\text{n : \(\mathbb{N}\)} \\
\text{lt\(_0\) n : 0 \leq n}
\end{array}
\]

\[
\begin{array}{c}
\text{n : \(\mathbb{N}\)} \\
\text{m : \(\mathbb{N}\)} \\
\text{p : n \leq m}
\end{array}
\]

Introduction Rules

\[
\begin{array}{c}
\text{Ex : lt\(_3\) (lt\(_3\) (lt\(_0\) 2)) : 2 \leq 4}
\end{array}
\]
Dependent Types as Fibrations

- Formation for vectors: \[ A : \text{Type} \quad \eta : \mathbb{IN} \]
  \[ \text{Vec}_{A \eta} : \text{Type} \]

Vec \(_B^0\) \quad Vec \(_B^1\) \quad Vec \(_B^2\) \quad Vec \(_B^3\)

Vec \(_B^0\): [\(\emptyset\)]  
Vec \(_B^1\): [\(\top\)]  
Vec \(_B^2\): [\(\top, \top\)]  
Vec \(_B^3\): [\(\top, \top, \top\)]

Vec \(_B^1\): [\(F\)]  
Vec \(_B^2\): [\(T, T\)]  
Vec \(_B^3\): [\(T, F, T\)]

IN: 0 1 2 3 ...
Quantifiers

- Dependent Product (Forall)
  \[ A : \text{Type} \quad x : A \vdash B : \text{Type} \]
  \[ \prod_{x : A} B : \text{Type} \]

- Dependent Sum (Exists)
  \[ A : \text{Type} \quad x : A \vdash B : \text{Type} \]
  \[ \Sigma_{x : A} B : \text{Type} \]
"Geometric" Interpretation of Quantifiers

\[ \sum \quad \text{Total space} \]
\[ \Pi \quad \text{Space of sections} \]
Martin-Löf's Methodology

- **Formation** - In what context is a type well-formed?

- **Introduction** - How do I construct terms of the type?

- **Elimination** - How do I use the terms of my type?

- **Computation** - How do introduction and elimination interact?
Elimination + Computation

\[ \varphi : \Sigma B \quad \text{substituting } x : A \]

\[ \text{fst } \varphi : A \]

\[ \text{snd } \varphi : B[^{\text{fst } \varphi / x}] \]

\[ \text{fst } (a, b) = a \]

\[ \text{snd } (a, b) = b \]

\[ f : \Pi B \quad a : A \]

\[ \text{substituting } x : A \]

\[ f_a : B[^{a / x}] \]

\[ (\lambda x . b) a = b[^{a / x}] \]
Normalization and Canonicity

- The combination of these rules lets us reduce intro/elim pairs:

\[
\text{fst} \ (\text{snd} (\text{snd} (4, 7))) \ : \ \text{IN} \\
\text{refl} \ \text{fst} \ (4, 7) \ : \ \text{IN} \\
\text{refl} \ 4 \ : \ \text{IN}
\]

- This equips type theory with a notion of computation

- A meta-theorem (canonicity) asserts that all closed terms reduce to introduction forms.
Part II

Homotopy Theory
Martin-Löf Identity Types

\[ A : \text{Type} \quad a : A \quad b : A \]
\[ \text{Id}_A \quad a \quad b : \text{Type} \]

\[ a : A \]
\[ \text{refl} \ a : \text{Id}_A \ a \ a \]

- The only way to prove equality is reflexivity.
- This works modulo the computation rules
  \[ \text{refl} \ 4 : \text{Id}_{\mathbb{N}} \ 4 \ (3+1) \]
Curious Features

1. Because the formation rule is stated for any $A$, it can be iterated:

\[
\begin{align*}
\text{Id}_{A} & \quad \text{Id}_{A} \quad pb \quad \text{Id}_{A} \quad \text{Id}_{A} \quad pb \quad \text{Id}_{A} \quad \text{Id}_{A} \quad pb \quad \text{Id}_{A} \quad \text{Id}_{A} \quad pb \quad \text{Id}_{A} \quad \text{Id}_{A} \quad pb \quad \text{Id}_{A} \quad \text{Id}_{A} \quad pb
\end{align*}
\]

2. We cannot assume proofs of identity are unique.

What are we to make of this?
The Homotopy Interpretation

- Hoffman-Streicher (1994-95)

\[ \text{Axiom K: } \prod_x \prod_y \text{Id}_{A^{xy}} = \text{Id}_{A^{xy}} \]

is not provable.


Type theory can be interpreted in (certain) Quillen Model Categories

- Lumsdaine/Garner-Van der Berg (2008-9)

Types give rise to weak oo-groupoids
Groupoid Laws

- Can construct composition operation \( p \cdot q : \text{Id}_A \times 2 \)
- Can show various laws up to higher cells:

\[
\text{unit}_L : \text{Id}_A \times y \xrightarrow{p} P \quad (p \cdot \text{refl}_y)
\]

\[
\text{assoc} : \text{Id}_A \times x w \xRightarrow{(p \cdot q) \cdot r} (p \cdot (q \cdot r))
\]
Eckmann-Hilton

- Some laws are non-obvious and come directly from topology.

\[ \alpha \cdot \beta = \beta \cdot \alpha \]
Fibrations Revisited

\[ \text{transp} \, \mathcal{B}_{a_0} \, p : \mathcal{B}_{a_1} \]
**H-level**

- We can use identity types to stratify the universe
- First, define **contractible types**
  \[ \text{is-contr } X := \sum_{x : X} \prod_{y : X} \text{Id}_X^x y \]
- Now define **h-level** by induction
  \[ \text{has-level } (-2) X := \text{is-contr } X \]
  \[ \text{has-level } (S \ n) X := \prod_{x,y : X} \text{has-level } n \left( \text{Id}_X^x y \right) \]
Low Dimensions

-2 • Contractible types
  • It and only if equivalent to 1
  • Implies Identity types also contractible

-1 • Propositions
  • Types with “at most one” element
  • Play the role of truth values

0 • Sets
  • Elements are equal in at most one way
    \( \mathbb{N}, \mathbb{R}, \mathbb{Z}, \mathbb{B} \)

1 • Groupoids
  • Elements can have symmetrics FinType
Equivalences

- Homotopically correct notion of isomorphism

- Define the homotopy fiber of a map $f: X \to Y$

  $$\text{hfib} \ f \ y := \sum_{x: X} \text{Id}_Y \ (f \ x) \ y$$

- Say a map $f$ is an equivalence if all its homotopy fibers are contractible

  $$\text{is-equiv} \ f := \prod_{y: Y} \text{is-contr} \ (\text{hfib} \ f \ y)$$

- Being an equivalence is a proposition!

  $$\text{Equiv} \ A \ B := \sum_{f: A \to B} \text{is-equiv} \ f$$
Extensionality Principles

- One defect of Martin-Löf's identity type is that it fails to correctly reproduce the "natural" equality for some types.

- Function Extensionality

\[ \text{Id}_{A \to B} f g \equiv \prod_{a : A} \text{Id}_B (fa) (ga) \]

is \textcolor{red}{\text{not}} provable.

- It is often assumed as an axiom. But this breaks canonicity!
Univalence

- Another type which Martin-Löf's identity types fail to determine is \( \text{Type} \).
- What is the natural notion here?
- Voevodsky:

\[
\text{Id}_{\text{Type}} A B \equiv \text{Equiv} A B
\]
Univalence and Paths

- We can use univalence to produce examples of equalities which are not themselves equal.

Univalence \implies \bot (Axiom K)

\[\begin{array}{c}
\text{Type} & \text{Id} & \text{B} & \text{B} \\
\hline
T & F & \text{Id} & \text{B} & \text{B} \\
F & T & \text{Id} & \text{B} & \text{B} \\
\end{array}\]
Higher Inductive Types

- Type theory has long struggled from the absence of a reasonable theory of quotients.

- Higher inductive types generalize inductive types by allowing introduction rules to return not only elements of the type being defined, but also its identity types.
Examples

- $S'$
  - base: $S'$
  - loop: $\text{Id}$ base base $S'$

- $T$
  - pt: $T$
  - $p: \text{Id}_T$ pt pt
  - $q: \text{Id}_T$ pt pt
  - $\alpha: \text{Id}_T (p \circ q) (q \circ p)$
  - $\text{Id}_T$ pt pt

Diagram:

- Loop $Q$
- Base $S'$
- Points $p$, $q$, $pt$
- Alpha $\alpha$
Results from Homotopy Theory

- Homotopy Groups \( \pi_n(S^n) = \mathbb{Z} \), \( \pi_3(S^2) = \mathbb{Z}/2 \)
- Fibration Sequences
  \[ F \rightarrow E \rightarrow B \rightarrow \cdots \rightarrow \pi_n(F) \rightarrow \pi_n(E) \rightarrow \pi_n(B) \rightarrow \pi_{n-1}(F) \rightarrow \cdots \]
- Eilenberg-MacLane Spaces (Cohomology)
- Spectral Sequences
- (Generalized) Blakers-Massey Theorem

\( \Rightarrow \) Freudenthal Suspension Thm
Cubical Type Theories

- Inspired by homotopy interpretation
- Extensionality principles provable!
- Native HITs
- Implementation in Agda
Thanks!