Foundations of Mathematics: A Crash Course in Type Theory

HoK Bachelor's Thesis

Dominik Kirst

Saarland University
Department of Philosophy

SAARLAND UNIVERSITY
COMPUTER SCIENCE

Foundations of Mathematics

Dependent Type Theory

Discussion

HISTORICAL SKETCH

Situation in the late 19th century:

- Mathematical branches based on their own assumptions
- Mostly following the *axiomatic method*:
  Purely logical derivation of theorems from first-principles
- Unifying approaches proposed by Cantor and Frege
- These systems were found to be logically inconsistent by Cantor, Burali-Forti, Zermelo, Russell, ...

Two solutions to the foundational crisis:

- Axiomatic Set Theory: ZFC (Zermelo, Fraenkel, ...)
- Dependent Type Theory: MLTT (Martin-Löf, ...)

[Zermelo, 1908], [Fraenkel, 1925], [Martin-Löf, 1985]

What is a Foundation of Mathematics?

Mathematics = Objects + Reasoning

A foundational system provides a common language and logical system for *all of* mathematics. Important properties are:

- Universality: every mathematical concept can be expressed and no particular branch is preferred
- Precision: the language is unambiguous and the assumptions and steps in every argument can be identified
- Effectiveness: there are simple algorithms checking the well-formedness of statements and correctness of proofs
- Consistency: the system is empirically free of logical flaws

What is a Satisfactory Foundation?

Mathematics = Objects + Reasoning

What are mathematical objects?
What are mathematical proofs?

A satisfactory foundation answers these and related questions in a convincing way. Furthermore, the system is *practical*:

- Accessibility: the system is simple and intuitive
- Mechanisability: proof checking and proof automation
- Community: improvement and standardisation

The Idea of Types

A type is a collection of objects sharing an operational property.

Typical examples:

- Natural numbers $n : \mathbb{N}$
- Numerical functions $f : \mathbb{N} \to \mathbb{N}$
- Functionals $F : (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N})$
- Boolean predicates $p : \mathbb{N} \to \mathbb{B}$

Typing rules fix the operational behaviour of the types.
Typeable statements: $(f\,n : \mathbb{N}), (F f : \mathbb{N} \to \mathbb{N}), (F f\,n : \mathbb{N}), (p\,n : \mathbb{B})$
Untypeable statements: $(n\,n : ?), (f\,f : ?), (F\,n : ?), (F\,p : ?)$

The idea of Computations

Types consist of canonical elements as well as *computations*.

Consider the identity $I : \mathbb{N} \to \mathbb{N}$ on natural numbers:

$$I\,n := n \qquad \rightsquigarrow \qquad I := \lambda n.\, n$$

Terms like $I\,3$ and $I\,n$ for $n : \mathbb{N}$ have type $\mathbb{N}$ and describe procedures that evaluate to canonical numbers. For instance:

$$I\,3 = (\lambda n.\, n)\,3 \succ 3 \qquad\qquad 3 + 2 \succ 4 + 1 \succ 5$$

Computation is well-behaved on typeable terms.

More Types

There are two primitive types:

- Unit type $\top$ with a single canonical element $T : \top$
- Empty type $\bot$ with no canonical element

We have seen the type former $A \to B$ for function types.
There are two more primitive type formers:

- Product type $A \times B$ of pairs (a, b) for $a : A$ and $b : B$
  with projections $p_1 : A \times B \to A$ and $p_2 : A \times B \to B$
- Sum type $A + B$ with elements $i_1\, a$ and $i_2\, b$
  for injections $i_1 : A \to A + B$ and $i_2 : B \to A + B$

Indexed Products and Sums

Using the natural numbers $\mathbb{N}$ we can think of a generalisation of the binary products and sums for types $A_0, A_1, A_2, \ldots$:

$$A_0 \times A_1 \times A_2 \times \ldots \qquad\qquad A_0 + A_1 + A_2 + \ldots$$

Members of the first: functions giving an element of A_n for $n : \mathbb{N}$
Members of the second: pairs of an index n and a member of A_n

So binary products and sums have *dependent* counterparts:

- $\Pi(x : B).\, A\, x$ containing functions $\lambda x.\, s$ with $(\lambda x.\, s)\, b : A\, b$
- $\Sigma(x : B).\, A\, x$ containing pairs (b, a) with $b : B$ and $a : A\, b$

Here $A : B \to U$ is a type family on a type U of types.

Internal Logic

Consider the typing rules for function application and pairing:

$$\frac{\vdash f : A \to B \qquad \vdash a : A}{\vdash f\,a : B} \qquad\qquad \frac{\vdash a : A \qquad \vdash b : B}{\vdash (a, b) : A \times B}$$

These have a well-known logical reading if we interpret $\to$ as implication, $\times$ as conjunction, and leave out the terms.

The internal logic can be summarised by two slogans: *Propositions-as-types (CH)* and *proofs-as-terms (BHK)*.

For instance, a logical formula ϕ corresponds to the type A of its proofs. Then the logical tautology that ϕ implies itself is reflected by the type $A \to A$. This type is inhabited by the term $\lambda a.\, a$ which corresponds to a proof of the implication.

Internal Logic: Overview

Logical concept	Interpretation	Proof term
Truth	$\top$	the canonical proof
Falsity	$\bot$	no proof
Conjunction	$A \times B$	pair of proofs of A and B
Disjunction	$A + B$	either a proof of A or of B
Implication	$A \to B$	function from A to B
Negation	$A \to \bot$	special case of the above
$\forall$-quantification	$\Pi(x : A).\, P\, x$	function mapping a to $P\, a$
$\exists$-quantification	$\Sigma(x : A).\, P\, x$	a witness a and a proof of $P\, a$

Equality on a type A is expressed by an additional type $s =_A t$:

$$\frac{\vdash s : A}{\vdash e\, s : s =_A s} \qquad \frac{\vdash H : P\, s \quad \vdash P : A \to U}{\vdash E_{=}\, H : s =_A t \to P\, t}$$

Some Peano Arithmetic

Natural numbers are generated from zero and its successors:

$$\frac{}{\vdash 0 : \mathbb{N}} \qquad \frac{}{\vdash S : \mathbb{N} \to \mathbb{N}} \qquad \frac{\vdash s : A \quad \vdash f : A \to A}{\vdash E_{\mathbb{N}}\, a f : \mathbb{N} \to A}$$

The eliminator $E_{\mathbb{N}}$ enables direct recursive definitions:

$$m + n := E_{\mathbb{N}}\, m\, (\lambda n'.\, S\, n')\, n \qquad m * n := E_{\mathbb{N}}\, 0\, (\lambda n'.\, m + n')\, n$$

These satisfy the desired equations *computationally*:

$$\begin{aligned} m + 0 &\succ m & m * 0 &\succ 0 \\ m + (S\, n) &\succ S\, (m + n) & m * (S\, n) &\succ m + (m * n) \end{aligned}$$

More Peano Arithmetic

Proving the Peano Axioms means to construct elements of the types corresponding to the respective statements:

- Disjointness: $\Pi n.\, 0 \neq S\, n$
- Injectivity: $\Pi m n.\, S\, m = S\, n \to m = n$
- Induction is established by a generalised eliminator:

$$\frac{\vdash s : P\,0 \qquad \vdash f : \Pi(n : \mathbb{N}).\, P\,n \to P\,(S\,n)}{\vdash E_{\mathbb{N}}\, s f : \Pi(n : \mathbb{N}).\, P\,n}$$

Here $P : \mathbb{N} \to U$ is interpreted as a unary predicate on $\mathbb{N}$.

Mathematical Foundation

- Universality: the elementary mathematical concepts are all primitive and the internal logic has the extend of high-order predicate logic
- Precision: the language is unambiguously formal and the internal notion of proof provides fully detailed reasoning
- Effectiveness: well-formedness and correctness of proofs are both instances of the type checking algorithm
- Consistency: MLTT can be proven to be equiconsistent to (strong and constructive versions of) ZFC set theory

Practicality

- Accessibility: MLTT similar to ZFC if presented in reasonable detail, captures the pre-formal intuition of mathematical types, concepts like induction an recursion built-in
- Mechanisability: easily implementable such as any other functional programming language, already led to groundbreaking and extensive formalisation projects (Four-colour theorem, Kepler conjecture, etc.)
- Community: currently ZFC is the mathematical mainstream, hence MLTT is less standardised and has to deal with suspicions

Convincing Answers

Mathematics = Objects + Reasoning

What are mathematical objects? - Mental constructions!

What are mathematical proofs? - Mental constructions!

These constructions are intuited by the human mind instead of presupposing the existence of platonic ideas as abstract entities in some conceptual universe. Truth means provability.

Intuitionistic Logic

Not every classical proof corresponds to a construction.
Hence MLTT naturally implements intuitionistic logic:

- Statements such as the *law of excluded middle* unprovable:

$$\Pi(A : U).\, A + \neg A$$

- However, these can still be assumed consistently
- Proofs of disjunctions and existentials bear information:

$$\Pi(x : A).\Sigma(y : B).\, P\,x\,y + Q\,x\,y$$

- No need for explicit computational models
- Advantageous for subclassical analyses

Higher-Order Logic

Objects of any order are accommodated as first-class entities. Hence MLTT naturally implements higher-order logic:

- Uniform quantification and function application:

$$\Sigma F.\,\Pi(A : U)(P : A \to U).\,P\,(F\,P)$$

- No *complete* proof system can exists (for full semantics)
- Completeness plays a minor role in a constructive setting
- Instead admits *categorical* descriptions of structures
- Internal functions in ZFC simulate some of this strength:

$$\exists f.\,\forall p \in \mathcal{P}(\omega).\,f(p) \in p$$

Equiconsistency of MLTT and ZFC

Types-as-sets interpretation:

- Type-theoretic constructions can directly be interpreted by their set-theoretic counterparts: $[\![A \times B]\!] := [\![A]\!] \times [\![B]\!]$, etc.
- Yields a denotation function $[\![\_]\!]$ from typed terms to sets such that $[\![a]\!] \in [\![A]\!]$ holds whenever $\vdash a : A$
- Hence every proof $c : \bot$ entails an element $[\![c]\!] \in \emptyset$

Sets-as-trees interpretation:

- The inductive type of well-founded trees can be shown to satisfy most set-theoretic axioms
- Hence every derivable contradiction within a certain subsystem of ZFC can already be simulated within MLTT

CONCLUSION

Both MLTT and ZFC are formal foundations of mathematics.

However, there are advantages of MLTT over ZFC:

- Suitability for modern applications
- Convincing philosophical system
- Informative intuitionistic logic
- Expressive higher-order logic
- Inductive types with built-in recursion

No definite answer but MLTT seems a good candidate for now.

References

Fraenkel, A. (1925).
Untersuchungen über die Grundlagen der Mengenlehre.
Mathematische Zeitschrift *22*, 250–273.

Martin-Löf, P. (1985).
Intuitionistic Type Theory: Notes by Giovanni Sambin of a Series of Lectures Given in Padua, June 1980.
Prometheus Books, Napoli.

Zermelo, E. (1908).
Neuer Beweis für die Möglichkeit einer Wohlordnung.
Mathematische Annalen *65*, 107–128.