Homotopy Type Theory

Reading Club (Summer 2025)

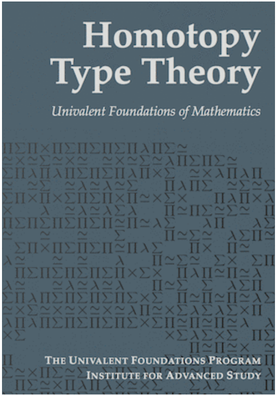
Robin Rawiel
robin.rawiel@uos.de
CV Group

Ulf Krumnack
krumnack@uos.de
AI Group

April 2025

Everything You Always Wanted to Know About Homotopy Type Theory*

- read the HoTT book
- meet and discuss with like-minded
- teeming terminology, fancy formulae, thrilling theorems, and categorical constructions
- have some fun time



*But Were Afraid to Ask

Roadmap

Homotopy Type Theory (HoTT)

What is Homotopy Type Theory?

- a relatively new branch of mathematics
  - to a certain extent work in progress
  - broad outlines have been fixed
- combines aspects of several different fields in a surprising way
- based on a recently discovered connection between homotopy theory and type theory
  - homotopy theory: an outgrowth of algebraic topology and homological algebra, with relationships to higher category theory
  - type theory: a branch of mathematical logic and theoretical computer science
- connections are currently the focus of intense investigation
  - "it is increasingly clear that they are just the beginning of a subject that will take more time and more hard work to fully understand"

Homotopy Type Theory

A short history of HoTT ...

- early 20th century: origins of type theory
  - developed by logicians like Bertrand Russell and Alonzo Church
  - designed to avoid certain paradoxes
- since 1970s: dependent type theory
  - proposed by Per Martin-Löf as foundation for computer science and constructive mathematics
  - applications in proof assistants
- early 2000s: exploring connections between type theory and homotopy theory
  - Vladimir Voevodsky: univalence axiom
  - matured through collaborative efforts
- Recent developments:
  - 2019: The International Conference on Homotopy Type Theory
  - 2023: The 2nd International Conference on Homotopy Type Theory

Origins of Type Theory

- Bertrand Russell introduce type theory to deal with paradoxes in the logical foundations of mathematics, like Russel's paradox (in naive set theory):
  - let R be the set of all sets that do not contain themselves.
  - is $R \in R$? $\rightarrow$ paradoxical situation
  - problem: unrestricted set formation
  - solution: organize sets (and propositions) into a hierarchy of types: a type can only contain elements of a lower type
- Alonzo Church developed a rigorous formal system in the context of his "λ-calculus"
  - a formalization of computation based on a simple calculus with only 3 rules: function abstraction, function application, and substitution
  - the untyped λ-calculus is Turing complete, but can lead to paradoxes
  - the simply typed λ-calculus avoids paradoxes (but is no longer Turing complete, as it restricts unbounded recursion).

Type Theory: How does it work?

- in type theory, unlike set theory, objects are classified using a primitive notion of type
  - similar to the data-types used in programming languages.
- elaborately structured types can be used to express detailed specifications of the objects classified
  - giving rise to principles of reasoning about these objects
- simple examples:
  - the objects of a product type $A \times B$ are known to be of the form (a, b), and so one automatically knows how to construct and decompose them
  - an object of function type $A \to B$ can be acquired from an object of type B parametrized by objects of type A, and can be evaluated at an argument of type A
- rigidly predictable behavior of all objects
  - extensive use in verifying the correctness of computer programs
  - basis of modern computer proof assistants: formalizing mathematics and verifying the correctness of formalized proofs

Propositions as type

- Curry-Howard isomorphisms:
  - logical proposition can be considered as types
  - proofs corresponds to "inhabitants" of types
- Example:
  - logical conjunction $A \wedge B$ corresponds to product type $A \times B$
  - to proof $A \wedge B$ we need a proof p for A and a proof q for B
  - in type theory: $p : A$ and $q : B$
  - the pair (p, q) has type $A \times B$ (is a proof for the proposition $A \wedge B$)
- Logical problems can be mapped to computational problems and vice versa
  - many central results discovered indepently can be reinterpreted
- proof verification can be viewed as type checking
  - type checking can be automatized
  - computer proof assistants

Dependent Type Theory

- Per Martin-Löf, among others, developed a “predicative” modification of Church’s type system
  - usually called *dependent*, *constructive*, *intuitionistic*, or simply *Martin-Löf type theory*.
- originally intended as a rigorous framework for the formalization of constructive mathematics
- basis of the system used in Homotopy Type Theory
- in this course: when saying “type theory” we usually refer specifically to this system
  - we will get (informally) introduced to that system in chapter 1 of the book
  - formal treatment in appendix A

Homotopy theory

What is homotopy theory?

- homotopy theory originated in algebraic topology:
  - concerned with spaces and continuous mappings between them, up to homotopy.
  - a homotopy between a pair of continuous maps f and g may be thought of as a "continuous deformation" of f into g
- nowadays homotopy theory is often done on a more abstract level
  - categorical treatment: Quillen model categories, n-categories
- spaces X and Y are said to be homotopy equivalent, $X \equiv Y$, if they are isomorphic "up to homotopy", i.e.
  - there are continuous maps going back and forth
  - the composites of which are homotopical to the respective identity mappings
- homotopy equivalent spaces
  - are said to have the same homotopy type
  - have the same algebraic invariants (e.g., homology, fundamental group, ...)

Homotopy Type theory

- make precise the basic concept of type: regard types
  - not as strange sets (perhaps constructed without using classical logic)
  - but as spaces, viewed from the perspective of homotopy theory
- solves the problem of understanding how the notion of equality of elements of a type differs from that of elements of a set.
- the "holy trinity" (Curry-Howard-Lambek correspondence) as an extension of the Curry-Howard correspondence:

$$\begin{tikzcd}
 & \text{Types} \arrow[dl, leftrightarrow] \arrow[dr, leftrightarrow] & \\
\text{Logic} \arrow[rr, leftrightarrow] & & \text{Categories}
\end{tikzcd}$$

  - in many situations there are correspondences between these theories
  - ideas can be viewed from different perspectives with new insights

The Role of Homotopy Type Theory

- bring new ideas into the very foundation of mathematics
- two central ingredients:
  - Voevodsky's subtle univalence axiom: isomorphic structures can be identified
  - higher inductive types: provide direct, logical descriptions of some of the basic spaces and constructions of homotopy theory: spheres, cylinders, truncations, localizations, etc.
- both are impossible to capture directly in classical set-theoretic foundations
- but when combined in homotopy type theory, they permit an entirely new kind of "logic of homotopy types"

Univalent Foundations

The Univalent Foundations program:

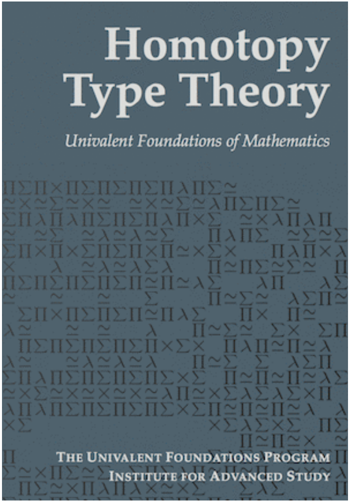
- HoTT suggests a new conception of foundations of mathematics
  - often referred to as "univalent foundation" with reference to the central "univalence axiom"
  - an alternative to Zermelo-Fraenkel set theory (ZFC)
- allows for convenient machine implementations
  - can serve as a practical aid to the working mathematician

From the preface of the book:

> *[...] we therefore believe that univalent foundations will eventually become a viable alternative to set theory as the "implicit foundation" for the unformalized mathematics done by most mathematicians.*

Roadmap

Content of the book



- collective work by the *Univalent Foundations Program*
- output of a special year held in 2012-13 at the Institute for Advanced Study
- published in 2013
- electronic version downloadable for free
- printed version for reasonable price
- `https://homotopytypetheory.org/book/`

About the book

> *We did not set out to write a book. The present work has its origins in our collective attempts to develop a new style of "informal type theory" that can be read and understood by a human being, as a complement to a formal proof that can be checked by a machine. Univalent foundations is closely tied to the idea of a foundation of mathematics that can be implemented in a computer proof assistant. Although such a formalization is not part of this book, much of the material presented here was actually done first in the fully formalized setting inside a proof assistant, and only later "unformalized" to arrive at the presentation you find before you – a remarkable inversion of the usual state of affairs in formalized mathematics.*
>
> *from the "Preface"*

About the book

The book

- is intended as
  - a first systematic exposition of the basics of univalent foundations, and
  - a collection of examples of this new style of reasoning
- consists of two parts
  - Part I: Foundations
  - Part II: Mathematics
- does not require the reader to know or learn any formal logic, or to use any computer proof assistant
- should be regarded as a “snapshot” of the state of the field at the time it was written
  - is continuously updated (newer copies may include changes)

Why choosing this book?

Pros and cons:

- relatively new approach, only few alternative text books available
- reviews are overall positive
- written in an informal style
  - may be easier to follow as less distraction by formalities
- contains exercises
  - allows to assess understanding
- written by a collective of authors, may be less coherent
- written by mathematicians for mathematicians (not cognitive scientists)

Roadmap

How is this course organized?

- Lecturers:
  - Robin Rawiel
  - Ulf Krumnack
- Format: Seminar
  - the course is worth 4 ECTS
  - AI (bachelor/master)
- Weekly meetings (Thursday 16:00-18:00)
  - 50/E07
  - StudIP forum for discussions
- Intended audience:
  - master or advanced bachelor students

Motivation for the Course

Why do we offer this course?

- intrinsic interest in understanding the presented ideas
  - the relation of formal and informal mathematics, structure of (practical) mathematical reasoning
  - a more solid understandig of type theory
  - the role of homotopy theory
- the somewhat abstract and partly unfamiliar nature of the topics suggests active discussion
- potential applications in math and proof assistants
- requests by students for abstract/algebraic/categorical math/logic courses

Target Audience

Who should participate in this course?

- students with some background and/or vivid interest in logic, math, or type theory
- we do not expect any specific preknowledge
  - we will use our sessions to introduce and discuss unfamiliar ideas
  - only works with active participation!
- logic: some familiarity with basic logic concepts
- math: some idea of mathematical proofs
- type theory: introduced in the first chapter of the book
- homotopy theory: introduced in the second chapter of the book

Tasks and Responsibilities

What to do to get a certificate for this course?

During the week:

- **obligatory**: read the text ($\sim$ 1 chapter per week)
- **obligatory**: participate in forum discussion
- optional: do some exercises
- optional: do some background research

In our Tuesday meetings:

- **obligatory**: participate in the discussion
- **alternating**: chair the session
- optional: present exercises
- optional: short background presentations

Chairing a Session

Tasks for the chair:

- develop an agenda for the meeting
- decide which parts of the text need special attention
- include points that came up in forum discussion
- moderate the discussion

Choosing a chair:

- the next chair will be decided at the end of each session
- chairing may be done in teams

Chairing a Session

Non-tasks for a chair:

- the chair **should not** act as a solo entertainer
  $\rightarrow$ they should rather involve everybody into the discussion
- the chair **should not** give a detailed presentation of the text
  $\rightarrow$ we assume that everybody has read the text
  $\rightarrow$ the chair may show the text to structure the session
- the chair **should not** summarize and present the forum discussion
  $\rightarrow$ they should rather engage the original contributor to explain their points

Roadmap

Next Steps

Until next week: prepare chapter 1: “Type theory”

- read the chapter (up and including section 1.9)
- post in the StudIP forum:
    - state central concepts with short explanation
    - clarification questions: try to state what exactly is unclear
    - answer questions from other participants
    - comment on the text: what did you learn, what is interesting, what seems pointless, ...
    - ideas, cross-connections, references, etc.
    - exercises: highlight interesting/instructive exercises (from the end of the chapter or external sources)
- the earlier the better (give others a chance to react!)
    - latest by Sunday evening

Next Steps

Until next week: prepare chapter 1: “Type theory”

- read the chapter (up and including section 1.9)
- post in the StudIP forum:
  - state central concepts with short explanation
  - clarification questions: try to state what exactly is unclear
  - answer questions from other participants
  - comment on the text: what did you learn, what is interesting, what seems pointless, ...
  - ideas, cross-connections, references, etc.
  - exercises: highlight interesting/instructive exercises (from the end of the chapter or external sources)
- the earlier the better (give others a chance to react!)
  - latest by Sunday evening

Chair for next session: anybody volunteering?

Thank you!

Questions?