Cubical Type Theory

Theory of Truth, Types, Equality, Identity, Universes, Univalence, HITs, Kan Filling, Cubical Type Theory

The related trinity of type theory, mathematical logic/proof theory and category theory forms a modern foundation for all of mathematics and computation.

A type is a precise mathematical specification of behavior. An element of a type satisfies its specification. Both a type and its elements are programs, subject to evaluation. Elements evaluate down to values and types evaluate down to canonical types [Martin-Löf] or type values [Harper] (same idea). Application developers experienced with mainstream programming languages will already understand elements (objects) are subject to evaluation, but will find types being so to be new and significantly more expressive. Type theory supports the idea of indexed families of types-the index being a value (also known as dependent types). This will also be a new idea to most developers. By introducing these ideas and lots more, modern type theory goes far beyond what we see as type systems for regular programming languages. This course focuses on the most promising of the latest type theories, known as cubical type theory- a computation-friendly approach to new ideas such as univalence & higher inductive types. The redtt open source project is a good implementation.

Target Audience

This course is aimed at modern developers who need a better grasp of how areas of mathematics can be practically applied to programming.

Prerequisites

Good foundational mathematical education along with some programming experience, as we include exploring type theory from a computational viewpoint. Attendees can select which programming language they wish to use, as all concepts will be developed from first principles.

Contents of One-Day Training Course

A New Foundation

Type theory
Category theory
Mathematical logic/proof theory
How they relate to each other (Shulman)
Constructivism

Type Theory Overview

Think of modern type theory as a highly expressive programming language useful for mathematics (+everything based on it)
A theory of truth – based on Martin-Löf's ground-breaking paper: "Constructive mathematics and computer programming"

Basic Concepts

Type
Element
Evaluation
Judgment
Witness
Intuitionistic Type Theory

Simple Types

Defining a type
Defining an element
Look at bool and nat(ural number)

Common Types

Type
Function
Sum
Product
Top / Bottom

Dependent Types

The idea behind dependent types
Creating an indexed families of types

Additional Concepts

The meaning explanation
Functionality (in a type theory sense)
The hypothetical
Recursion

Equality and Identity

Introduction to equality in type theory
Variations of equality type (exact, ..)
Identifications

Higher Inductive Types (HITs)

HITs as generalization of inductive types
Uses in various forms of construction
Diagonals

Univalence

Initially considered in terms of Homotopy Type Theory (HoTT)
That uses an axiom – not desirable from a computational viewpoint – why?
So other approaches explored ...

Cubical Type Theory - Intro

This is the cutting edge (2019) type theory
Simple intro
Uses points, lines, planes, cube, n-cube
The idea of paths
Transporting

Cubical Type Theory - Details

Kan filling
Higher dimensionality
Cartesian variant

Redtt

Redtt is an open source implementation of cubical type theory – let's explore how it works