15-814 Types and Programming Languages
Introduction and Overview

Frank Pfenning

Department of Computer Science
Carnegie Mellon University

Tue Sep 1, 2020

Outline

Course Details

- Lectures and attendance, recordings [Canvas > Zoom]
- Class communication [Piazza]
- Lecture notes and (optional) textbook [Web Pages]
- Weekly homeworks (solo, unless noted o.w.) [Canvas]
- Midterm & final exams [?]
- Grades: 60% Homework, 15% Midterm, 25% Final [?]
- Software [`linux.andrew.cmu.edu`]

Why Do We Study Programming Languages?

- Programming is at the heart of computer science
- Languages are not "all the same"
  - Some languages are measurably better than others
  - Languages have different purposes
- Fundamental concepts transcend individual languages

How Do We Study Programming Languages?

- Break down into small universal building blocks (e.g., functions or pairs)
- Types are the central organizing principle
- Focus on semantics, not syntax
  - How does it compute (operational)
  - What does it compute (logical)
- Investigate properties of all programs expressible in a language

The λ-Calculus

1901 Russell's Paradox in naive set theory

1910 Russell & Whitehead's Ramified Theory of Types

1933 Church tries to replace sets by functions
(but: inconsistent)

1936 The λ-Calculus (Church & Rosser)

1936 Turing Machines, the Church-Turing Thesis

1940 Church's Simple Theory of Types
(now: consistent)

1980 Martin-Löf's Intuitionistic Theory of Types

2020 15-814!