OREGON PROGRAMMING LANGUAGES SUMMER SCHOOL

Lambda-Calculi for Logics — Valeria De Paiva

Lecture 1 - *June 23, 2025*

1 History

At the start of the 20th century (c. 1920-1930), a math revolution took place. Mathematics shifted from a focus on numbers themselves to a focus on algebra and proofs - the structures connecting these numbers. However, it was a "silent" revolution since mathematics didn't undergo as explosive of a change as other fields like biology (the theory of evolution) or physics (the quantum revolution). Math is often still perceived as nothing more than its pre-revolution contents, but the concepts introduced in this period of change are incredibly relevant to computing today. Here are some of these relevant concepts:

- Category theory
  - further organizes how concepts and structures relate to one another
- Curry-Howard correspondence precursors
  - 1847: Boolean algebras
  - 1920: Schonfinkel conbinatory logic
  - 1945: Kleene realizability

$$
\begin{tikzcd}[column sep=small]
 & \text{Church } \lambda\text{-Calculus} \arrow[dl, dash] \arrow[dr, dash] & \\
\text{Cartesian Closed Category} \arrow[rr, dash] & & \text{Intuitionistic Prop. Logic}
\end{tikzcd}
$$

The Curry-Howard correspondence links logics, programming languages and categories. (Initially, it only connected logics and programming languages.) It came about because Hilbert wished to formalize consistency of arithmetic using finitistic methods in such a manner that the formalism was

1. Consistent - no contradictions
2. Complete - all true statements are provable
3. Conservative - results about "real" objects don't need to rely on "ideal" objects
4. Decidable - statements can actually be shown to be T or F

Gödel's Incompleteness Theorems (1931) showed that this list was not possible with current methods. Gentzen, Hilbert's student, sought to find more powerful methods. Some mathematical highlights of this wartime period include:

- Gödel (1933, 1942, 1958) created a liberalized version of Hilbert's program to justify classical systems as intuitively as possible.
- Gentzen invented systems of natural deduction and sequent calculus to prove consistency of arithmetic
- Church's lambda calculus (1936) and Church Thesis that λ-definability, recursive functions, and Turing machines are equivalent

Curry-Howard for Implication

- We connect proofs and programs by using the Curry-Howard correspondence:

- Natural deduction without λ-term

$$\frac{A \to B \qquad A}{B} \qquad\qquad \frac{\begin{array}{c}[A] \\ \vdots \pi \\ B\end{array}}{A \to B}$$

- Natural deduction with λ-term

$$\frac{M \;:\; A \to B \qquad N \;:\; A}{M(N) \;:\; B} \qquad\qquad \frac{\begin{array}{c}[x \;:\; A] \\ \vdots \pi \\ M \;:\; B\end{array}}{\lambda x : A.\, M \;:\; A \to B}$$

This led to lambda calculus as a universal programming language, which in turn led to new logics and constructs to extend it. Category theory is helpful for finding and reasoning about these extensions.

2 Category theory

Category theory is the general mathematical theory of structures and systems.

- **Types:** formulae/objects in a category
- **Terms/programs:** proofs/morphisms in an *appropriate* category (the appropriate part is the hard part)
- **Category:** collection of objects & morphisms, where morphisms are arrows between objects that satisfy identity and associativity properties
- **Functor:** morphisms between categories
- **Natural transformation:** morphisms between functors
- **Reduction is proof normalization** (Tait)

Why categories?

- Prioritizes modeling derivations rather than truth/falsity - these derivations are useful in many fields (linguistics, functional programming, compilers)
- Allows you to solve a problem where it's easier, then transport the solution

- Can be used with many different logics:
  - System F
  - Classical logic
  - Intuitionistic logic
  - Linear logic
  - Dependent type theory
  - Modal logic
  - High-order logic, etc.