CS4860 Lecture 26 (Final Lecture) Type Theory

Fall 2012 Nov. 29, 2012

Computer science is concerned with type theory as well as set theory because types are used to structure thinking about programming languages. Algol 60, Algol 68, and Pascal are programming languages with rich type systems. In 1972 C.A.R. Hoare wrote a very influential paper, Notes on data structuring, which presented a mathematical basis for reasoning about data types.

At the same time that set theory was being axiomatized by Zermelo, Russell axiomatized his theory of types which formed the framework for the book Principia Mathematica by Whitehead and Russell, 1910 for Vol I of three volumes. This work probably influenced the presentation of types in programming languages. Type theory went out of fashion in mathematics because set theory is easier to axiomatize. But Russell's definition that a type is the range of significance of a function remained as a strong organizing principle for both programming languages and mathematical foundations based on functions (rather than sets alone) as primitives.

The most fundamental point about type theory whether for programming or constructive mathematics, is that it is based on a computation system over a type of (syntactic) terms.

In this lecture I summarized the computation system for Heyting Arithmetic (HA). The material is covered in the supplemental reading, The Structure of Nuprl's Type Theory, pages 20-23.