The Curry Howard Isomorphism Summary

Gregory Malecha

February 16, 2008

The book develops type theory and in particular focuses on its strong connection to proofs. In the presentation, I will focus on Chapter 4, "Introduction to Type Theory". The talk will be based predominantly upon examples and exercises from the text and the correspondence between programs and proof will be stressed in the relationship among the various rules.

This chapter develops the concepts behind basic types starting with the propositional calculus ($\neg$, $\vee$, $\wedge$, and $\Rightarrow$) and their connections with sum, product and arrows types. Types are defined in terms of 4 rules: *formation*, *introduction*, *elimination*, and *computation*. Formation rules give a structural definition of types. Introduction and elimination rules show how the types interact with syntactic constructs of the programming language. The computation rule gives small-step semantics for the constructs introduced for the types. Examples such as transitivity of implication are given, proven, and their connections to the λ-calculus are shown.

From the foundations, the book grows the system to include both universal and existential quantification and shows how they are provide abstraction at the level of types. The connection between the universal quantifier and implication, and between the existential quantifier and the product type are shown, stressing the significance of constructive in their formulation. Numerous examples and exercises are given for these, some of which will be covered in the presentation.

Next base types including the booleans, finite (enumerated) types, and the natural numbers are introduced. Upon these foundations, equality and inequality are introduced and are used to build dependent types.