

A Denotational Semantics for STLC

D. Welch

Abstract. Just some notes trying out “ \LaTeX literate Agda mode” as I work through this next chapter on the lambda calculus (eventually denotational semantics for the untyped version of the calculus).

1 Intro to the Lambda Calculus

The lambda calculus, first published by Alonzo Church in 1932 is a minimal calculus with only three syntactic constructs: variables, abstraction, and (function) application.

The *simply typed lambda calculus* (STLC) is a variant of the lambda calculus published in 1940 that adds static typing to the o.g. 1932 untyped lambda calculus. It has the three types of constructions previously mentioned plus additional syntax to support types and type annotations. These notes use a ‘Programmable Computable Function’ (PCF) style syntax and add operations for naturals and recursive function definitions.

Specifically these notes formalize the base constructs that make up the simply-typed lambda calculus: its syntax, small-step semantics, and typing rules. After this a number of properties the language such as progress and preservation are stated and proven. The notes may extend the language with additional features such as records.

Note: these notes do not present a recommended approach to formalization as they (initially) eschew a locally nameless representation of terms done for example via DeBruijn indices. A later section of these notes might look into this.

1.1 Imports

First, we’ll need some imports: