

Grammar Lexical Analysis: Extracting Tokens to Form a DFA

Tyler Rimaldi ^{*†}

Abstract

A Grammar specifies a rigid rule set for generating valid strings from a specific language's alphabet. A Grammar enables a program to transform source code (that is syntactically valid, according to the Grammar), that would normally act as a linear sequence of some format of characters, into a syntax tree, or a parse tree, through a process called Lexical Analysis. Lexical Analysis of a grammar breaks down a sequence of characters into a collection of tokens. This process assigns some predetermined meaning to each token thereby generating a deterministic finite automaton as a state machine (or regular expression). In this project, the first stage of a compiler – the Lexical Analysis phase – will be developed to generate a deterministic finite automaton based on a specified grammar. If time permits, further stages and enhancements not explicitly recognized in this abstract will be made as additions to the project with the consent of Professor Rivas.

^{*}Thank you to Prof. Rivas for the motivation for this project and for the opportunity to develop a state machine.

[†]Thank you to Prof. Labouseur for allowing me to use his grammar and for the project idea.