**CS323 Documentation**

About 2-3 pages

1. **Problem Statement**

*Make a parser that will print to an output file the tokens, lexemes and the production rules used; that is, first, write the token and lexeme found, then print out all production rules used for analyzing this token. This parser is to use the lexical analyzer made for project 1 to get the tokens and lexeme needed for the syntax analyzer to build a parse tree.*

1. **How to use your program**

*Cd to the directory called SA, there should be a file called “test”. Run “test” by typing ./test in the terminal.*

1. **Design of your program**

*A major component of our program is the lexical analyzer, that is the top part of our source code that analyzes the source code char by char and returns a token and stores a lexeme in a variable to be used by the syntax analyzer. The rules in the second half of the program which are for the syntax analyzer are also important. Each function is a nonterminal symbol that makes calls to other nonterminals until we reach an accepting state after CC() gets called. If we can’t reach an accepting state due to a syntax error we display an error message. The starting symbol is APrime() which we got by augmenting the grammar. This function only serves to call A() and return true to the if statement in main if the source code was successfully parsed indicating that that the parsing was completed successfully.*

1. **Any Limitation**

*There are no limitations to the size of the source code besides any standard limits that are set by ifstream. There are also no limitations within the code as far as we know.*

1. **Any shortcomings**

*<Anything you could NOT implement although that is required by the*

*Assignment.* ***Say ‘None’ if there is no shortcoming****>*