**CS323 Documentation**

About 2 pages

1. **Problem Statement**

*<write the problem statement here. You can mostly get it from the*

*assignment itself>*

Our goal is to build a syntax analyzer that is able to read when the code file has a grammar issue or if syntactically the code is correct and follows all the rules stated/given. We must be able to remove any left recursion and backtracking that may occur. Furthermore, the parser must hae the ability to print the tokens, lexemes, and production rules used. First it must print out the token and lexeme found and promptly following that it must print out the rules that it utilized to exist/continue. The parser must be capable of handling errors syntactically I.E if an error occurs on a line it must be able to print out a meaningful message such as the line number, error type, token, and lexeme.

1. **How to use your program**

*<write detailed steps how to execute your program>*

In order to utilize our syntax analyzer you will be required to run the executable file with “./parser” then the program will begin to run. On the terminal the question will arise as to which input test case file you’d like to test with your options ranging from test1- test3. These are predefined test case files. However, if you are to add your own all test case files must come with the postfix “.txt”. Input the desired text file into the command line and press enter. From there, all the tokens and their respective lexeme values with the rules utilized for them will be displayed inside the terminal window. If there are errors the parser will display a meaningful error message and cut the program early. Comments and string literals will be ignored by the syntax analyzer.

1. **Design of your program**

*< write major components of your program. Also, data structures you are utilizing, particular algorithms you have chosen etc. >*

1. **Any Limitation**

*<All features are running according to the assignment but you limit your program due to resource limitations, such as*

*Maximum number of lines in the source code, size of the identifier, integer etc.* ***Say ‘None’ if there is no limitation****>*

None

1. **Any shortcomings**

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

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

None