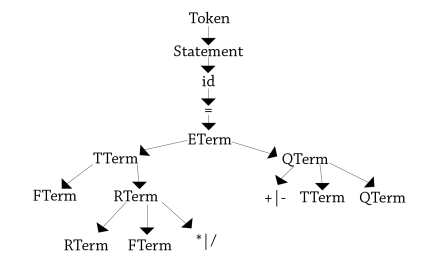
**Syntax Analyzer Documentation**

**1. Problem Statement**

The second assignment is to create a Syntax Analyzer that will take tokens from an input.txt file and use a set of production rules to determine if the statement is using the correct syntax. The output generated will show if the statement has been parsed successfully or not and show the production rules used to parse the statement.

**2. Design of your program**

This program uses a stack to keep track of the rules being used to parse the statements. It also keeps another stack that contains the input. As items are added to the input stack, production rules are applied to the syntax stack. Once a semi colon is added to the input stack the final operations are applied to the syntax stack. If by the end of the statement the syntax stack does not contain any values the statement has been successfully parsed. If the input stack contains any input that can not be identified from the rules required by the assignment, then the statement can not be parsed.



**3. Any Limitation**

The program assumes that the first token is an identifier. For example, it starts off with the production rule S 🡪 id = E. If the first token is not a identifier, the code will not identify the statement as valid.

There is a very minute issue when the input.txt file has an extra space at the end, the program will still compile and produce output, but the program will be stalled. Removing any extra spaces after the semi-colon will fix this minor bug.

**4. Any shortcomings**

None