**Peanut Lexer - CS323 Documentation**

1. **Problem Statement**

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

*assignment itself>*

1. **Usage**

There are two methods to use this program.

***Method 1 - Drag-and-drop:***

1. Using the File Explorer, navigate to the directory that *lexer.exe* is located in.

2. Drag-and-drop your Rat23F program (*.rat* file) onto the executable.

A computer screen shot of a computer

Description automatically generated

3. Open the generated output file to see the results.

A screenshot of a computer

Description automatically generated

***Method 2 – Command line:***

1. Using the Command Prompt, navigate to the directory that *lexer.exe* is located in.

2. Run *lexer.exe* while passing the path to the Rat23F source file to process as the first argument.

3. See the output in the console or open the generated output file to see the results.

A screenshot of a computer program

Description automatically generated

1. **Design of your program**

*Identifier*, *Real*, and *Integer* token types are each represented by respective finite state machines. The symbol set is composed of possible categorizations of source code characters: *Letter*, *Digit,* *Period,* and *Special*. When a finite state machine receives one of these symbols, its next state is looked up from its transition table, a 2D array of integers. Each state is represented by an integer that corresponds to the index of a row in the table. The 0th row is reserved as an “unrecoverable” state that is reached when an expected symbol is received at an unexpected time. When this occurs, or an unexpected symbol is received that results in undefined behavior, the finite state machine halts, and reports whether or not it is in an accepting state. If it halted in an accepting state, the string it iterated through is labeled according to its token type.

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****>*

1. **Any shortcomings**

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

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