**CPSC 323 Documentation**

**Problem Statement**

The problem statement for this assignment is to write a lexical analyzer (lexer) that can run "Rat21SU", a conventional and easy-to-understand programming language. The goal is to integrate a Finite Set Machine (FSM) into our lexer. This program can read a file containing the source code in Rat21SU and generate tokens and write out the results to an output file.

**How to Use the Program**

The project is written in C++. Below are the instructions on how to execute the program.

1. Download the compressed folder on your computer to the location of your choice.
2. Locate the downloaded file and extract the compressed folder.
3. Open the Linux Terminal.
4. Navigate to the directory where the files are located.
5. Navigate to the inside of the “.exe” directory.
6. Make sure that the three test case files are in the same directory.
7. Run the .exe file from the terminal.

**Design of your program**

Our program has six lexical units which are, identifiers, keywords, integers, operators, separators, and unknown (for error input).

We designed the DFSMs of identifiers and integers and created arrays for keywords, operators, and separators. The program is read character by character. When a string is read, we compare whether it is in the operators' array or the separators' array. Afterward, we compare the string to the DFSMs we created. If it is an identifier, then we check whether it is a keyword. If it is not a keyword or does not match anything listed in our array, the word "unknown" is printed due to error input.

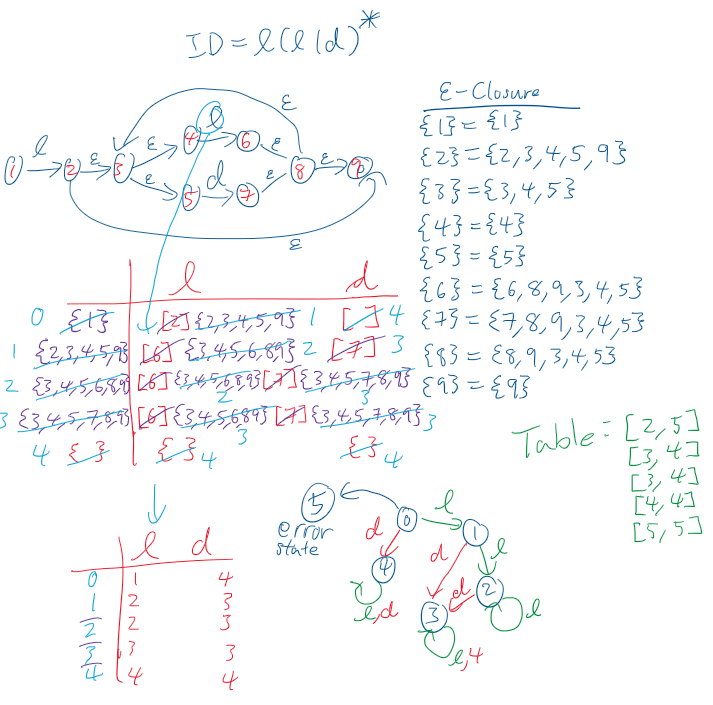
We created three test cases in which each one is unique because it depends on the number of lines of code. In the first test case, it is for ten or fewer lines of code. The second test case is for twenty or fewer lines of code. Lastly, the third test case is for twenty or more lines of code.

1. FSM of Identifiers
   1. Regular Expression:

l(l|d)\*

* 1. DFSM

|  |  |  |
| --- | --- | --- |
|  | l | d |
| q0=0 | 1 | 4 |
| 1 | 2 | 3 |
| 2 | 2 | 3 |
| 3 | 3 | 3 |
| 4 | 4 | 4 |

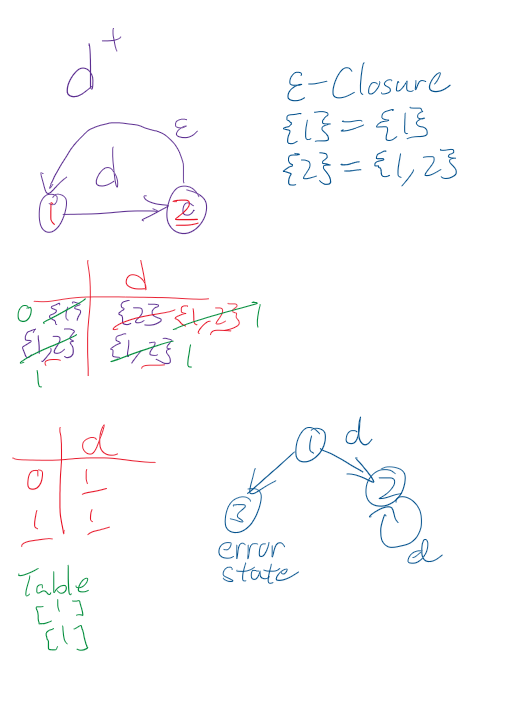


1. FSM for integers
   1. Regular Expression

d+

* 1. DFSM

|  |  |
| --- | --- |
|  | d |
| q0=0 | 1 |
| 1 | 1 |



**Any Limitation**

None

**Any shortcomings**

None