

CS323 Documentation

About 2 pages

1. Problem Statement

Our project seeks to address the issue of constructing a lexical analyzer for the programming language, RAT24S. The process of compiling a program begins with lexical analysis, which employs a lexical analyzer, AKA, lexer. The lexer has the function of dividing the source code into their token categories and identifying their respective lexemes. This is a multi-step process which begins with identifying the regular expressions for the tokens, utilizing Thompson's method to create an accompanying NFSM, then converting the NFSM to a DFSM through the subset method, and establishing a transition table for possible inputs, states, and accepting states. In implementing these structures into our program, we allow the lexer function to call the DFSMs of integer, int, and reals, which return whether the inputted token is valid or not. This allows the lexer to then identify the token and its corresponding lexeme. In addition, the established keywords, separators, and operators are identified and processed, and written to an output file, along with the results from the DFSM processing. The study and practice of lexical analysis is significant as it is the first step of the compilation process and understanding the breakdown of the process of identifying and validating tokens establishes a foundation upon which the rest of the understanding and implementation of the compilation process is built. The objective of this assignment is to gain a better understanding as to the mechanisms of a lexical analyzer deployed during the compilation process. By manually implementing one, it allows for further understanding of how lexers process input, segment the input, and call the corresponding FSMs to validate and identify them.

2. How to use your program

- a. *First make sure the main.py plus all the test case files are in the same directory.*
- b. *open up terminal in that directory, type: python main.py*
- c. *it should export the output into a text file called 'output.txt', read that to acquire the output.*

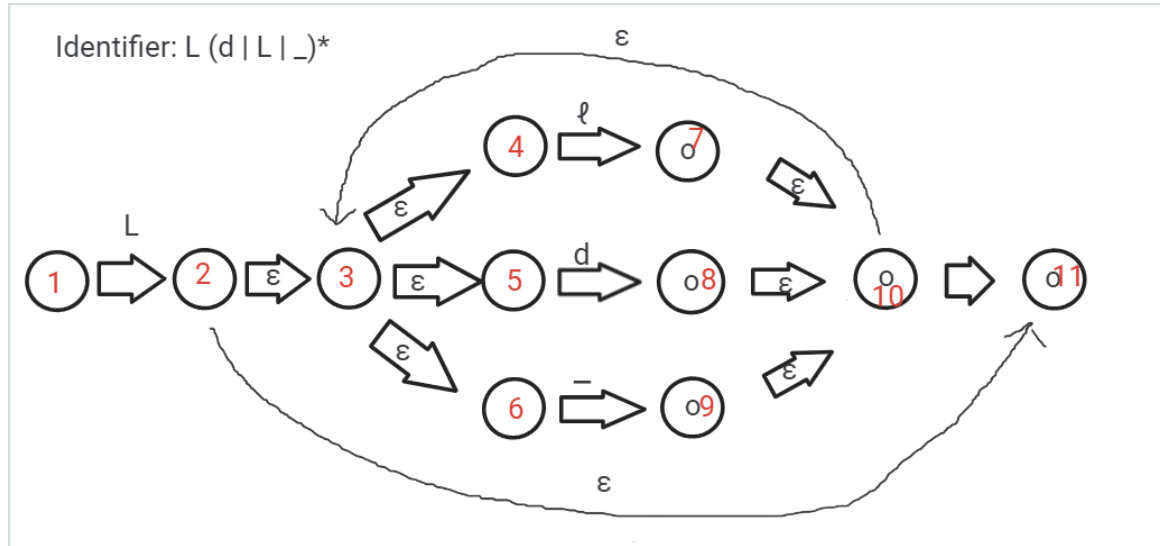
- d. you can modify the input of the test case by directly modifying the text file itself or replacing it with your own file with the same file name.
- e. The python version we are using is python 3.12

3. Design of your program

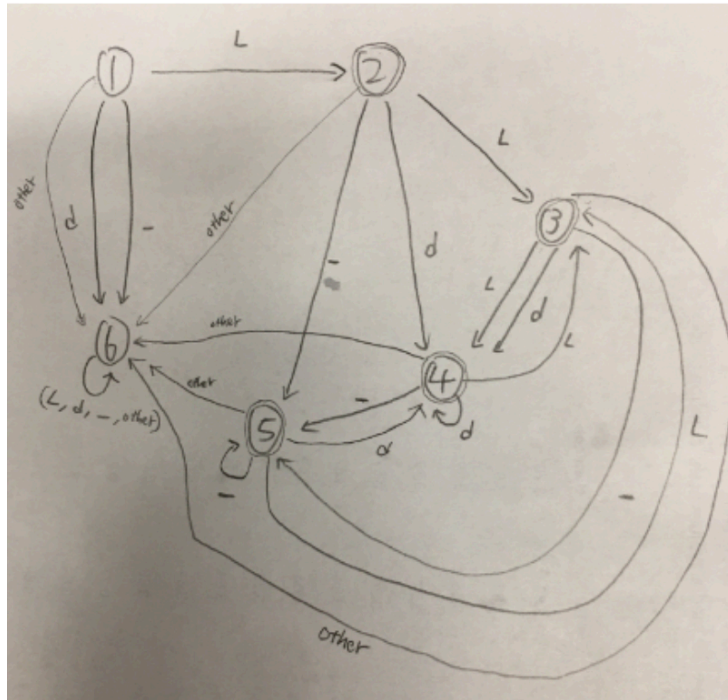
FSM Identifier

Regular Expression: $l(l|d|_)*$

NFSM Thompson Diagram:



DFSM:

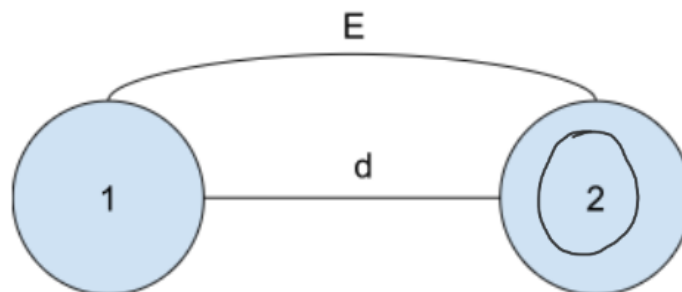


FSM Integer

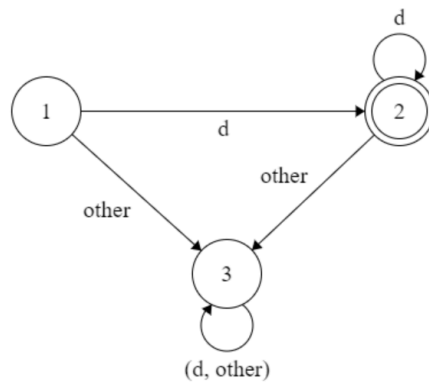
Regular Expression: d^+

NFSM Thompson Diagram:

Integer d^+



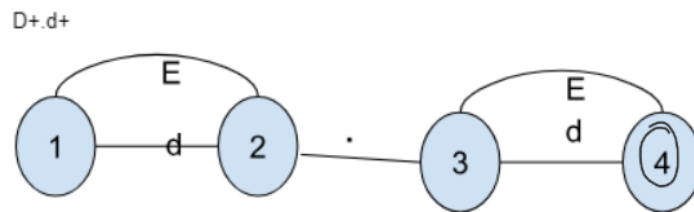
DFSM:



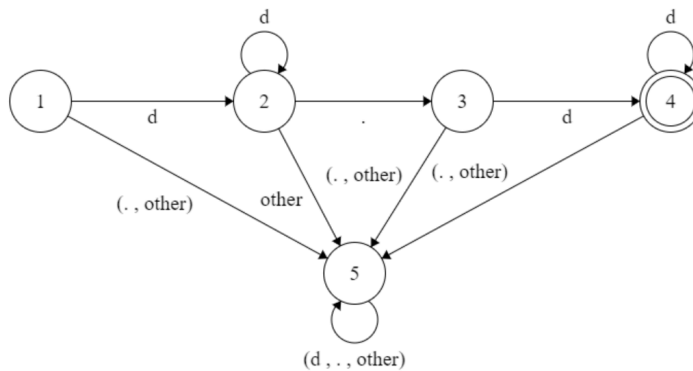
FSM Real

Regular Expression: $d+.d+$

NFSM Thompson Diagram:



DFSM:



Major components of our program include a lexer function, three DFSM functions for identifier, integer, and reals, as well as a function to remove comments, lists for separators, operators, and keywords, and a main function that holds three transition tables for identifier,

integer, and reals. Data structures utilized include Python dictionaries and lists.

4. 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

5. Any shortcomings

*<Anything you could NOT implement although that is required by the Assignment. **Say 'None' if there is no shortcoming**>*

None