Name: Aimen Khalid
Roll Number: BSCS19005

Compiler Construction
Assignment 1 - Report
Lexical Analyzer for PSL

- Lexer for PSL has been written in C++.
- PSL has some defined tokens: identifiers, keywords, comments, single line strings, multi line strings, float, int with specifications mentioned in the assignment document.
- DFAs for each of these tokens were constructed (submitted with the assignment).
- Using the DFAs, a single transition table was made on Excel and downloaded as a csv file.
- The rows of the table were possible input symbols and the columns were possible states that were defined in DFAs.
- There were a total of 81 input symbols and 58 states.
- Through this csv file, a 2D array named TT (short for Transition Table) was populated.
- A translation function was written to translate the input character from the row of the Transition Table to a row index of TT.

Algorithm:

The following algorithm was then used for the lexer function:

```
currState = q0
do{
        Read a character i from input file
        currState = TT[i][currState]
        if delimiter encountered:
                if currState is accepting state:
                        Identify the token and output <token, lexeme>
        If currState == errorState
                Report error
}while (end of file is not encountered)
```

Running the lexer:

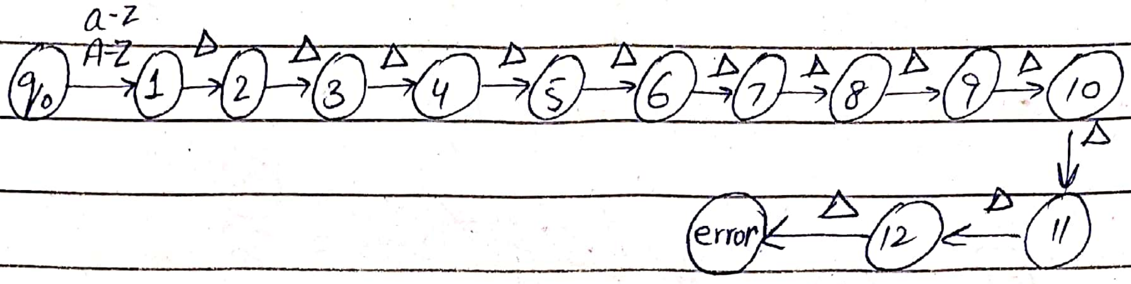
- To run the lexer, make sure the csv file is included in the project with the name "TT2.csv".
- The code on which lexer has to run should be written in file named "Text.txt". This file should be included in the project.

Aimen Khalid
bscs19005

Compiler Construction- Assignment 1

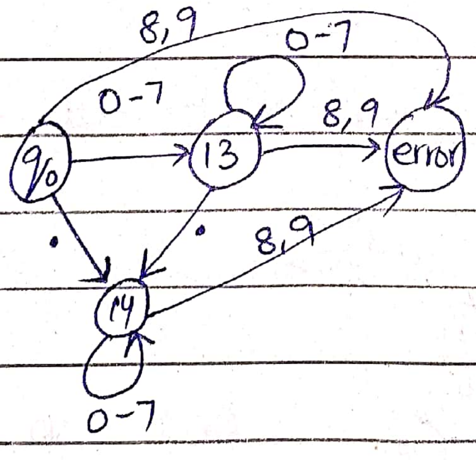
Δ = 0-9, a-z, A-Z, _

DFA for identifiers:



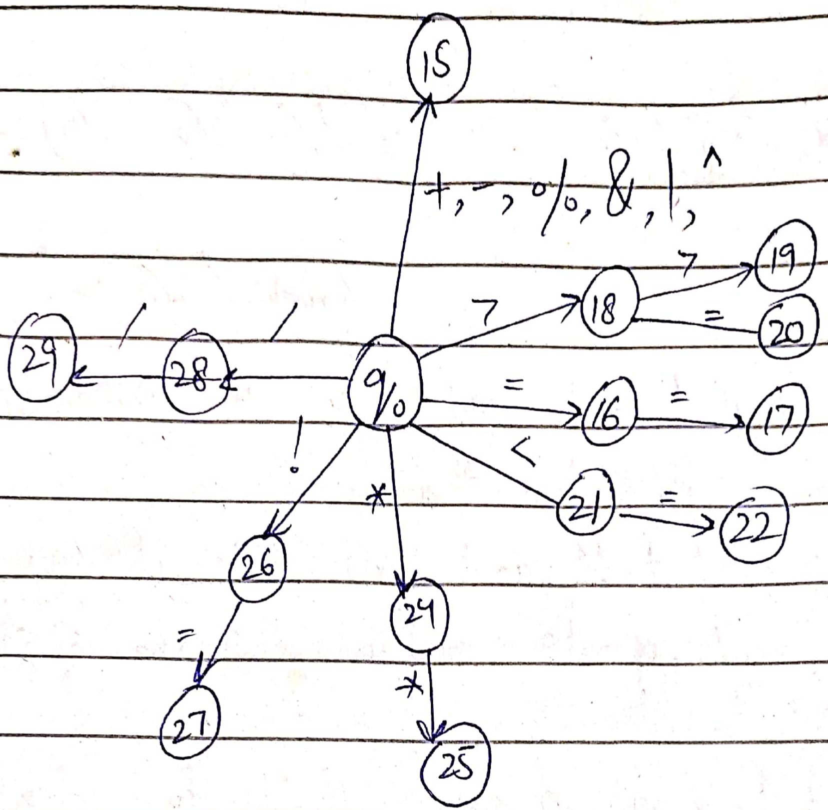
- States from 1 to 12 go to q0 if white space is given as input to them.
- States from 1 to 11 go to error state if any symbol other than Δ, operators, and white spaces is given as input to them
- States 1 to 12 are accepting states for identifiers.
- Identifiers are first checked for keywords.

DFA for int and float:

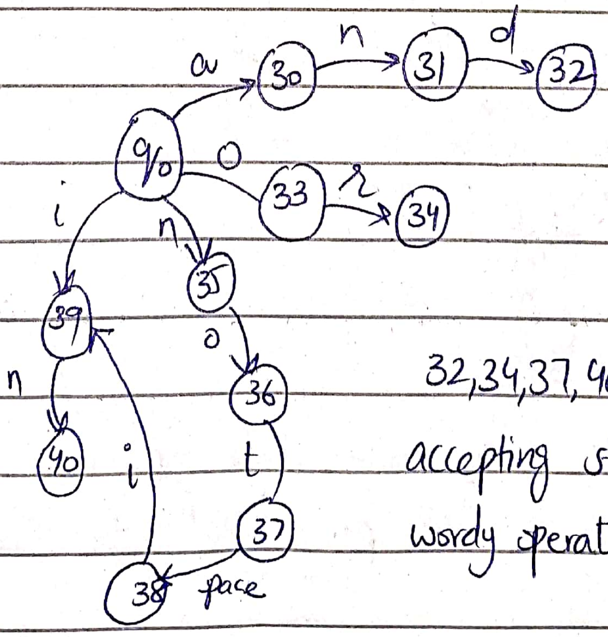


- State 13 is accepting state for int.
- State 14 is accepting state for float.

DFA for operators :

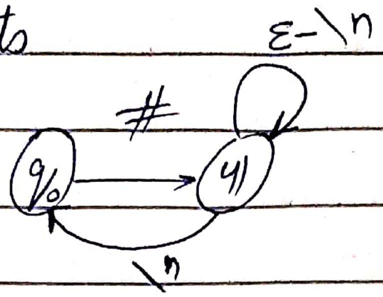


- All states accept 26 are accepting states for operators.

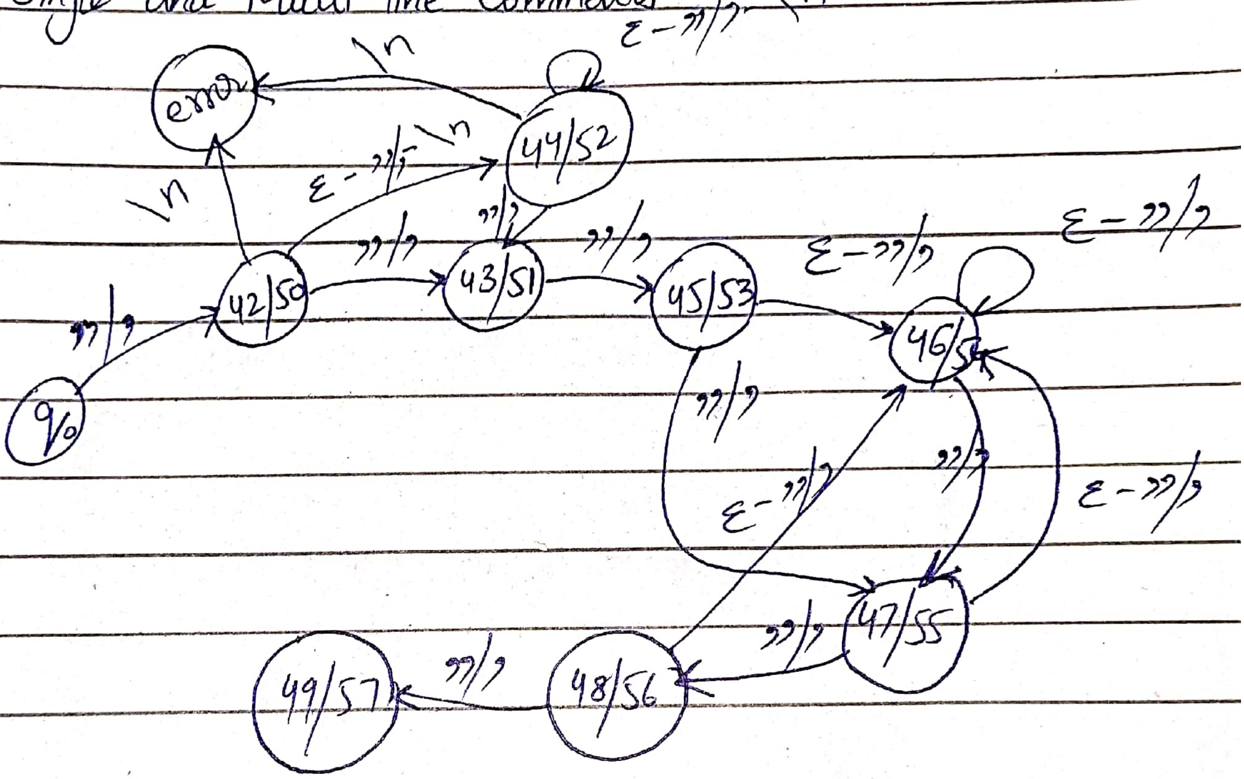


32, 34, 37, 40 are accepting states for wordy operators

Comments



Single and Multi-line ~~Comments~~ strings



- 43/51 are accepting states for single line ~~comments~~ strings.
- 49/57 are accepting states for multiple line strings corresponding to " and ' respectively.

Note:
More details of DFA can be seen from transition table submitted with the assignment.