Language Tool for Regular Languages.

FOR RUNNING THIS PROJECT:In util.Config file change absolute path for file folder

RULES FOR NFA and DFA:

1)Each state would be started fr om letter 'q'

2)No empty state named just 'q'

3)Transition syntax: qInput, char > qOutput;

4)Multiple transition syntax: qInput, char > qOutput1, qOutput2;

5)Each transition would be finished with ';'

6)Between transitions and start/final states declaring would be '-'

7)Start state syntax: begin:qStart;

8)Final state(s) syntax: final = qFinal1, qFinal2;

9)Empty input declared with dollar sign: '$'

RULES FOR REGULAR EXPRESSION:

1)If you want just single letter reg exp, write this char without any brackets: 'a'

2)Selection or Concat operation will be declared with brackets: (a|b) and (a.b)

3)Concat: with using dot '.'

4)Selection: with using '|'

5)Kleene star with ''*

5)Kleene star of regexp example : ((regexp))*

6)Kleene star of single char example : (a)*

7)Multiple operations will be divided by brackets

8)Valid examples: ((a|b)|c) or (a|(b|c) or ((a|b)|(c|d))

9)Valid examples: ((a.b).c) or (a.(b.c) or ((a.b).(c.d))

10)InValid examples: (a|b|c) or (a.b.c) or (a|b.c)