Documentatie Lab4 Week 6 LFTC

Socaciu Mihai 938

**Github link:** <https://github.com/Socca98/LFTC-2020>

Lab4 – Finite Automata

**Text file**

Q = { p, q, r }  
E = { 0, 1 }  
q0 = { p }  
F = { r }  
D = {  
 (p 0) - q,  
 (q 0) - r,  
 (r 0) - r,  
 (p 1) - p,  
 (q 1) - p,  
 (r 1) - r  
}

**1.** **Finite Automata Specification**

finite\_automata =

"Q = {" states "}"

"E = {" alphabet "}"

"q0 = {" initial\_state "}"

"F = {" final\_states "}"

"D = { \n" transitions "\n }"

states = state {“, ” state}

alphabet = alpha {“, ” alpha}

initial\_state = state

final\_states = state {“, ” state}

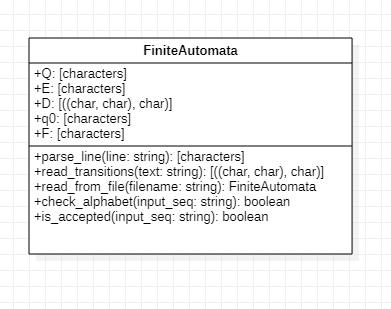
transitions = transition {“,” transition} \n

state = alpha {alphanumeric}

transition = "(" state " " alpha ") - " state

alphanumeric = “0” | .. | “9” | alpha

alpha = “a” | .. | “z” | “A” | .. | “Z”



def parse\_line(line):  
 *"""  
 Read elements between curly brackets and return them in lists.* ***:param*** *line: string, current line from text file* ***:return****: list of chars  
 """*

def read\_transitions(text):  
 *"""  
 Read the transitions (D) of the FA from file.* ***:param*** *text: string from file between '{}'* ***:return****: list of tuples  
 """*

def read\_from\_file(filename):  
 *"""  
 Read the FA from file and store it in an object.  
 Q - list of characters  
 A - list of characters  
 D - list of tuples (('p', '0'), 'q')  
 F - list of characters* ***:param*** *filename: Text file from which we read.* ***:return****: A new FiniteAutomata object storing multiple lists.  
 """*

def check\_alphabet(self, input\_sequence):  
 *"""  
 Check if some string contains only characters from the FA alphabet.* ***:param*** *input\_sequence: String of chars* ***:return****: True if ok, False otherwise  
 """*

def is\_accepted(self, input\_sequence):  
 *"""  
 Check if a sequence is accepted by the FA.* ***:param*** *input\_sequence:* ***:return****:  
 """*