Github source code: https://github.com/nicolagutanu/flcd/tree/main/carina/lab4

Class FiniteAutomata:

- finite_states = list
- alphabet = list
- initial_state = string
- final_states = list
- transitions = list

function read_from_file(self, file):
    input: file: string -> file name
    output: none
Reads the file line by line and places the data in the attributes of the FiniteAutomata class. The finite states, alphabet and final states as a list of strings, the initial state as a string as it can only be one and the transitions as a list of lists of a pair (current state, alphabet) and a string (the next state). (ex: [[('q0', '0'), 'q0'], [('q0', '1'), 'q1']])

function display_finite_states(self):
    input: none
    output: none
function display_alphabet(self):
    input: none
    output: none
function display_initial_state(self):
    input: none
    output: none
function display_final_states(self):
    input: none
    output: none
function display_transitions(self):
    input: none
    output: none

Prints the attributes of the FA (finite states, alphabet, initial state, final states and transitions) according to a menu.
ex:
Q={ q0,q1,q2 }
Σ={ 0,1 }
Initial state: q0
F={ q0 }
δ(q0 , 0)= q0

def check_dfa(fa, dfa):
	input: fa: FiniteAutomata (with its attributes)
		dfa: string -> sequence to be checked
	output: True/False
Checks if the given sequence is accepted by the FA. Starts at the initial state of the FA and then takes the sequence character by character and checks if the current state and the sequence character are in the transitions until it has reached the end of the sequence. If the sequence is correct, then the current state should have reached one of the possible final states and the number of transitions it's been through should be equal to the length of the sequence.

EBNF notation for the input file (FA.in):

letter = 'A'|'B'|...|'Z'|'a'|'b'|...|'z'
zerodigit = '1'|...|'9'
digit = '0'|zerodigit
state = letter | letter{ letter | digit }
finite_states = {state," "}
alphabet = {(letter|digit)," "}
initial_state = state
final_states = {state," "}
transitions = {state,",",(letter|digit),"=",state,"\n"}