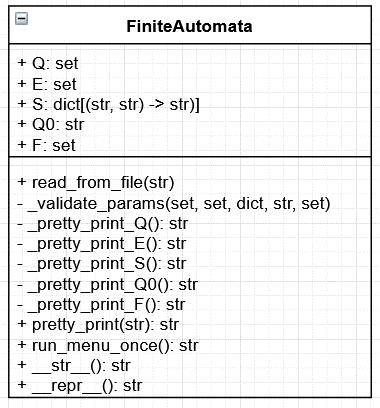
FLCD – Lab 4 – Documentation

Lung Alin-Sebastian

Gr. 934/2

Source code: <https://github.com/IcerOut/FLCD/tree/master/FiniteAutomata>

Application Design:



Implementation details:

The Finite Automata contains a set Q containing the states, a set E containing the alphabet, a dictionary S of form { (state ∈ Q, symbol ∈ E) -> state ∈ Q } containing the transitions, a string Q0 containing the initial state and a set F containing the final states

The structure of the input file is as follows:

p,q,r  
0,1  
p,0,q; p,1,p; q,0,r; q,1,p; r,0,r; r,1,r  
p  
p,r

The 1st line contains the states in Q, separated by comma

The 2nd line contains the symbols in E, separated by comma

The 3rd line contains the transitions separated by semicolon and space, with each element of the transition separated by comma

The 4th line contains the Q0 state

The 5th line contains the states in F, separated by comma

Q = Symbol { “,” Symbol }

E = Symbol { “,” Symbol }

S = Triplet { “; “ Triplet}

Triplet = Symbol “,” Symbol “,” Symbol

Q0 = Symbol

F = Symbol { “,” Symbol }

Symbol = { “a” | “b” | … | “z” | “A” | “B” | … | “Z” | “0” | “1” | … | “9” }

Q, E and F are stored as sets, so their uniqueness is guaranteed.

The pairs (state ∈ Q, symbol ∈ E) from transitions are used as dictionary keys, so their uniqueness is also guaranteed

The validate\_params(Q, E, S, Q0, F) function checks that all transitions and symbols only contain letters and numbers, that there is a single initial state, that each element used in the triplets of S is part of the respective set and that the final states ∈ Q

Test cases:

Input: “FA.in”

The output can be checked using the menu obtained by running “FA main.py”