Documentation

Github link: https://github.com/AmaliaDuma/Formal-Languages-and-Compiler-Design/tree/main/Labs/Lab_3%20-%20FA

class FiniteAutomaton:

- readFromFile(self, filename): Reads the elements of the fa from the file given as parameter

- isAccepted(self, sequence): Checks if sequence is accepted by parsing through the symbols in the sequence and: if we can't get a next state from crt symbol return false, otherwise continue. At the end we verify if the last state we obtained is a final state. If yes -> sequence accepted, otherwise -> sequence not accepted.

- _getNextState(self, state, value): Gets next state reachable from the crt state and the value we want to read

- print:

  - States, Alphabet, InitialState, FinalStates, Transitions: used for pretty printing of the elements on the screen.

Structure of fa.in file:

letter = "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"

digit = "0" | "1" | ... | "9"

symbol = letter | digit

state = letter {symbol}

states = state { "," state }

alphabet = symbol { "," symbol }

initState = state

finalStates = states

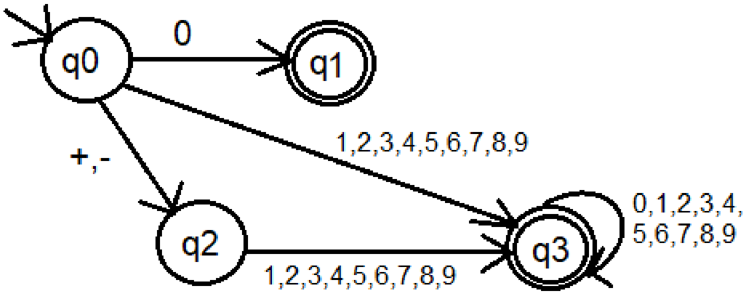
transition = state "," state "," symbol {"," symbol} transitions = transition {"\n" transition}

inputFile = states "\n" alphabet "\n" initState "\n" finalStates "\n" transitions

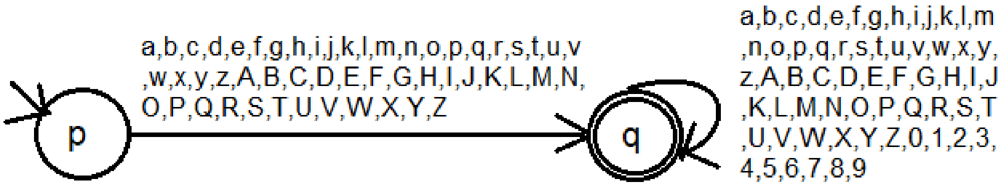
For grade 10:

Scanner now has 2 FA's which will read from the file fa_intConst.in and fa_identifier.in. Instead of matching a regex, for identifiers and int const scanner will check if the sequence is accepted by the 2 FA mentioned above.

fa_intConst.in ->



fa_identifier.in ->


p
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v
,w,x,y,z,A,B,C,D,E,F,G,H,I,J,K,L,M,N,
O,P,Q,R,S,T,U,V,W,X,Y,Z
q
a,b,c,d,e,f,g,h,i,j,k,l,m
,n,o,p,q,r,s,t,u,v,w,x,y,
z,A,B,C,D,E,F,G,H,I,J
,K,L,M,N,O,P,Q,R,S,T
,U,V,W,X,Y,Z,0,1,2,3,
4,5,6,7,8,9