Lab 4 Documentation

https://github.com/andreicupes/FLCD/tree/main/Lab2

We have added a new class which we have called "Finite Automaton". We create fields for the initial state, final states, all states, alphabet and transitions. The initial state is of type string, while the final states, all states and alphabet are a set of strings each one of them. The transitions will be saved in a map having as keys pairs of string values, while the value for each key will be a set of strings for the destination states. For a DFA, the set will be reduced to a single string value.

To check that our Finite Automaton is a DFA we check if the size of the set in the transitions is smaller or equal to 1. Moreover, to check if a given sequence is accepted by the DFA, we go through the characters 1 by 1, checking for each one if we can map it to a set with 1 value. If we cannot do that, we stop and return false. We return true if at the end of the sequence the last state is a final state.

In order to make it work with my previous code, I have replaced the regex expressions in the code with some given DFA's.

- The fa.in file is from where we will read our Finite Automaton.

digit ::= '0' | '1' |'2' | '3' |'4' | '5' |'6' | '7' | '8' |'9'

letter ::= 'A' | 'B' | …. | 'Z' | 'a' | 'b' | … | 'z'

alphabetChar ::= digit|letter

state ::= letter

transition :: state alphabet state

states ::= {state}+

alphabet ::= { alphabetChar }+

initialState ::= state

finalStates ::= {state}+

FAfile :: states '\n' alphabet '\n' initialState '\n' finalStates

- identDFA.in

digit ::= '0' | '1' |'2' | '3' |'4' | '5' |'6' | '7' | '8' |'9'

letter ::= 'A' | 'B' | …. | 'Z' | 'a' | 'b' | … | 'z'

identifier ::= letter {(digit|letter|_)}

- intDFA.in

sign ::= '+'|'-'

digit ::= '0' | '1' |'2' | '3' |'4' | '5' |'6' | '7' | '8' |'9'

digitno0 ::= '1' |'2' | '3' |'4' | '5' |'6' | '7' | '8' |'9'

intConst ::= [sign] digitno0 {digit}