

Finite Automaton

Source code: https://github.com/SerbanIoana/FormalLanguages-CompilerDesign/tree/main/lab6/lab4_FA

Classes

FiniteAutomaton – represents the FA

Representation:

List<String>: allStates

List<String>: alphabet

String: initialState

List<String>: finalStates

List<Transition>: transitions

Methods:

1. getFromFile()

Gets as input a file name and reads all the elements of the FA from the file.

2. verifySequence()

Gets as input a sequence and verifies if it is accepted by the FA. Parsing through the characters of the sequence and starting from the initial state, the method searches for any transition from the current state through the route specified by the current character. If no such route is found, it returns false. Otherwise, it continues until the sequence is done and checks if the current state is among the FA's final states, in which case the sequence is accepted.

3. getters and setters for all the elements of the FA

Transition – represents a transition from state1 to state2, on route

Representation:

String: state1

String: state2

String: route

Methods: getters and setters, toString

Main – contains the menu that will print all the information about the FA and a gives the user the possibility to specify a sequence that will be verified for acceptance against the FA

FA.in format

file := Q_list “\n” E_list “\n” q_0 “\n” F_list “\n” S_list

Q_list := “Q=” state {“, ” state} “\n”

E_list := “E=” symbol {“, ” symbol} “\n”

q_0 := “q0=” state “\n”

F_list := “F=” state {“, ” state} “\n”

S_list := “S=” transition {“, ” transition}

transition := “(“ state “,” symbol “,” state “)”

state := letter [letter | digit]

symbol := letter | digit

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

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