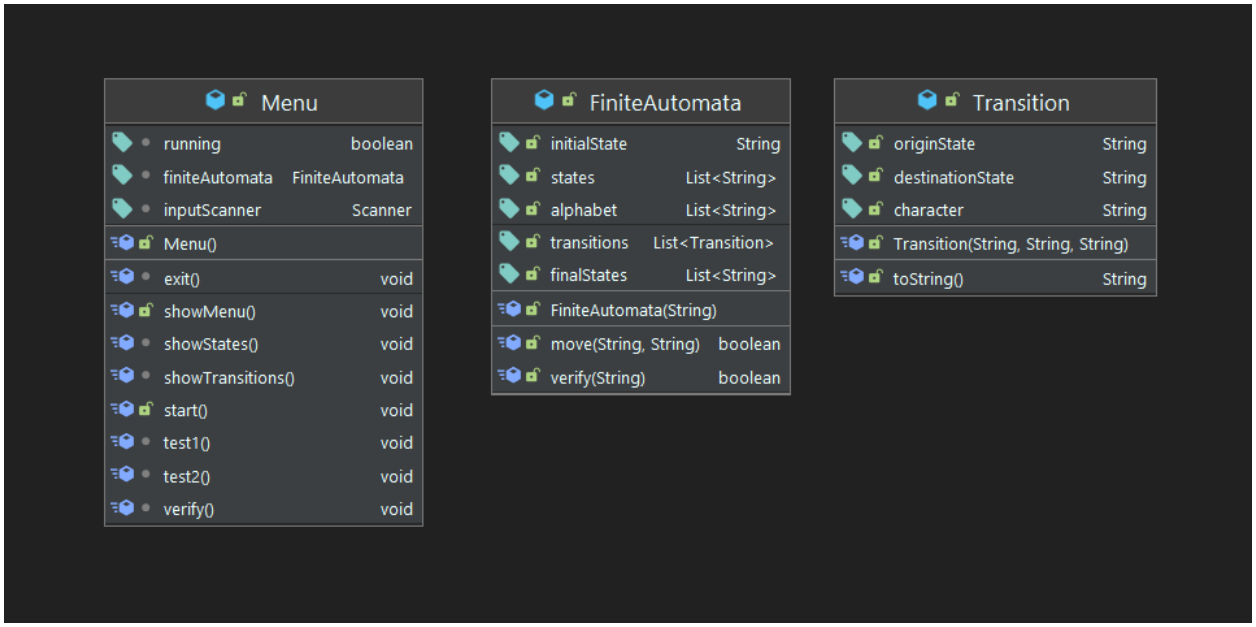
Lab 4

Requirements

Write a program that:

1. Reads the elements of a FA (from file)
2. Displays the elements of a finite automata, using a menu: the set of states, the alphabet, all the transitions, the set of final states.
3. For a DFA, verify if a sequence is accepted by the FA.

Architecture



Menu class

start()

- Start the main menu loop

showMenu()

- Show the menu options

showStates()

- Prints the FA states to the console

showTransitions()

- Prints the FA transitions to the console

test1()

- Verifies the sequence “aabc”

test2()

- Verifies the sequence “aba”

verify()

- Verifies a custom sequence from the system input

exit()

- Stops the main menu loop

FiniteAutomata class

FiniteAutomata(String filePath)

- Create a finite automata based on the specifications inside an input file
- Throws IOException on invalid input file
    - Duplicate states are not allowed
    - Duplicate alphabet entries are not allowed
    - There cannot be no transitions
    - Transition origin state is not within the defined states
    - Transition destination state is not within the defined states
    - Transition character is not within the alphabet
    - There cannot be no final states
    - Final state is not within the defined states

verify(String sequence)

- Verifies whether a sequence is accepted

move(String state, String sequence)

- Applies all transitions originating from “state” by recursively calling itself
- Returns true when a final state is reached with an empty sequence or when at least one move from the set of recursively called moves returns true

Design Details

FA.in specification:

alpha ::= 0 | ... | 9 | a | ... | z | A | ... | Z
digit ::= 0 | ... | 9

word ::= alpha{alpha}
number ::= number{number}
transition ::= word word word

states ::= number {word}
alphabet ::= number {word}
transitions ::= number {transitions}
finalStates ::= number {word}

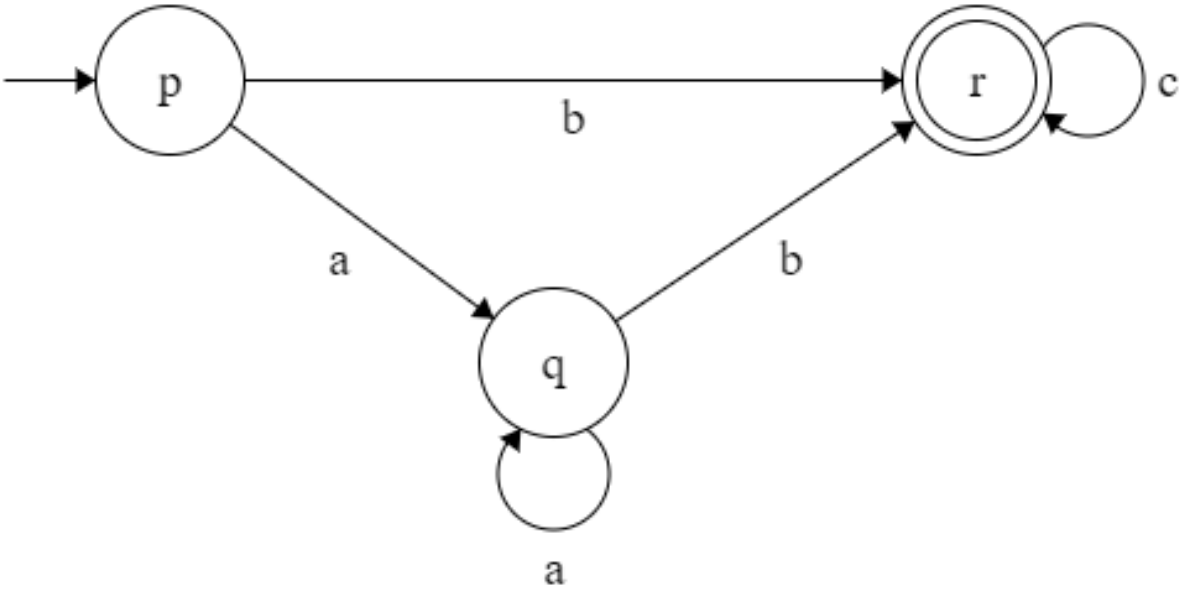
input ::= states alphabet transitions finalStates

Implementation

Github url: https://github.com/darius-calugar/flcd-fa-2

Tests

Tests were ran on the following finite automata:



Test 1

Sequence: **“aabc”**
Expected result: **“Accepted”**

Test 2

Sequence: **“aba”**
Expected result: **“Not accepted”**

Test 3

Sequence: **“”**
Expected result: **“Not accepted”**