Documentation for Lab4

LINK TO GIT: https://github.com/radutalaviniaelena/FLCD

REQUIREMENT:

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.

Deliverables:

1. FA.in - input file (*on Github*)
2. Source code (*on Github*)
3. Documentation. It should also include in BNF or EBNF format the form in which the FA.in file should be written (*on Moodle and Github*)

For implementing the laboratory, I have the following:

1) **Pair** class – is a template class that has three fields of type T1. I will use this class in order to keep a list of pairs, T1 representing the string element.

2) **Main** class – the most important functions are:

a) /**
* This function reads from a file all the values for states, final states and transitions, populating specific fields with them.
* **@param** filePath: a string value representing the location of the file
*/

public static void readFromFile(String filePath)

b) /**
* This function searches for the second state from a transition.
* **@param** firstState: a string value representing the first state from a transition
* **@param** value: a string value representing the value of a transition
* **@return:** the second search from a transition or “” in case it does not exist
*/
public static String getSecondStateBy(String firstState, String value)

c) /**
* This function verifies if a given sequence is accepted by the FA.
It is not accepted if -> 1. It does not start with the initial state
-> 2. It does not end with one of the final states.
-> 3. One of the values of the transitions does not exist.
* **@param** sequence: an array of string values
* **@return**: true, if the given sequence is accepted by the FA
false, otherwise
*/
public static boolean verifySequence(String[] sequence)

d) /**
* This function verifies if the given FA is deterministic or not. A FA is deterministic if there are different values for each first state of all transactions.
* **@return:** true, if the given FA is deterministic
false, otherwise
*/
public static boolean verifyIfDeterministicOrNot()

e) /**
* This function creates the menu of operations.
* **@return:** a string value representing the menu
*/
public static String showMenu()

f) **public static void main(String[] args)** – this function reads the file and treats all cases from the menu.

The EBNF of the input file (FA.in):

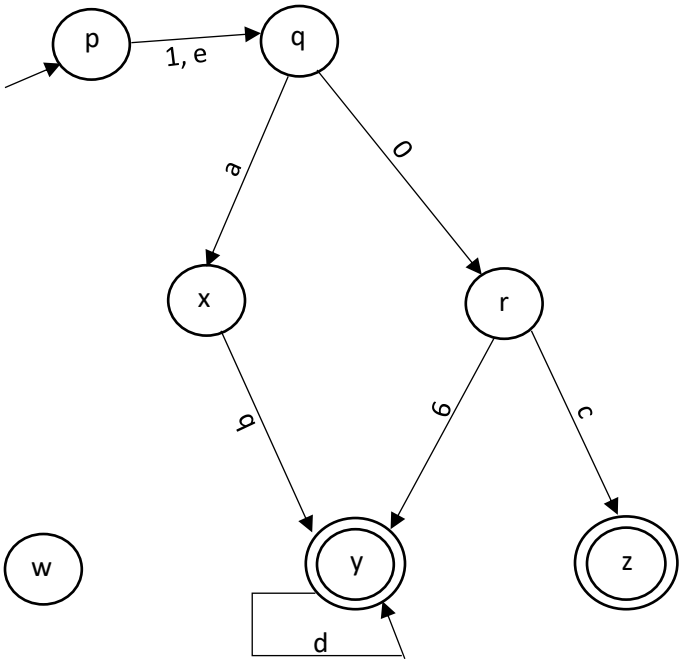
nz_digit := “1” | ”2” | .. | ”9”
digit := “0” | ”1” | ”2” | .. | ”9”
number := nz_digit {digit}

letter := “a” | “b” | .. | “z” | “A” | “B” | .. | “Z”
character := letter | digit
string := character {character}

first_line := number (* it represents the number of states *)
second_line := string {“ ” string} (* it represents the states of the DFA *)
third_line := string (* it represents the input state *)
fourth_line := number (* it represents the number of output states *)
fifth_line := string {“ “ string} (* it represents the output states *)
sixth_line := number (* it represents the number of transitions *)
seventh_line := {transition} (* it represents the transitions *)
transition := string string string

inputFile := first_line second_line third_line fourth_line fifth_line sixth_line seventh_line

Example:



FA.in | Main.java

```
7
p q x r w y z
p
2
y z
8
p q 1
p q e
q x a
q r 0
x y b
r y 6
y y d
r z c
```