https://github.com/Kun-Alexandru/UBB-projects/tree/main/SEM5/FLCD/Lab4

Finite Automata (FA) can be characterized as the simplest machine designed for pattern recognition. The organization of a Finite Automaton involves several key components:

Set of States (Q): Represented as an Array List of Strings.

Alphabet (Σ): Represented as an Array List of Strings.

Transitions (δ): Represented as an Array List of Transition objects.

Initial State (q): Represented as a String.

Set of Final States (F): Represented as a List of Array Strings.

Filename: Represents the file from which the data for the Finite Automaton will be read (as a String).

The organization of Transitions involves a startState (String), a value (String), and an endState (List of Strings).

Finite Automaton functionalities include:

Creating a new FA object by providing the filename of the file containing the necessary data.

Method readSet(String line, List<String> set): Splits the given line into tokens and adds the tokens to the provided set.

Method readTransition(String line): Splits the given line into tokens, creates a new Transition with those tokens, and adds the transition to the list of transitions.

Method readFromFile(): Reads the FA from the file specified by the filename given when the object was constructed.

Method isDFA(): Returns true if the FA is deterministic and false otherwise.

Method getNextState(String startState, String value): Iterates through all transitions with the given startState and value, returning the endState or "false" if there is no match.

Method isAccepted(String sequence): Checks if a given sequence is accepted by the FA. It goes from one state to another using the getNextState function and finally checks if the last state of the sequence is an endState from the set of final states of the FA.

Method toString(): Overrides the toString function to return the FA as a String.

BNF rules for FA input file:

<letter> ::= a | b | ... | z | A | B | ... | Z

<digit> ::= 0 | 1 | ... | 9

<enter> ::= \n

<setOfStates> ::= <letter> | <letter>,<setOfStates>

<alphabet> ::= <digit> | <digit> ,<alphabet>

<transition> ::= <letter>,<digit>,<letter> | <letter>,<digit>,<letter>,<setOfStates>

<setOfTransitions> ::= <transition> | <transition><enter><setOfTransitions>

<FA> ::= <setOfStates><enter><alphabet><enter><letter><enter><setOfStates><enter><setOfTransitions>