[FLCD/lab4/src/FA at main · 915-Negrila-Iulia/FLCD (github.com)](https://github.com/915-Negrila-Iulia/FLCD/tree/main/lab4/src/FA)

Finite Automata:

- defining FA:

- the simplest machine to recognize patterns

-FA organisation:

- contains objects:

1. (Q) Set of States as Array List of Strings

2. (Σ) Alphabet as Array List of Strings

3. (δ) Transitions as Array List of Transitions

4. (q) Initial State as String

5. (F) Set of Final States as List of Array Strings

6. Filename of the file from which the data for FA will be read (as String)

-Transition organization:

- contains startState(String), value(String), endState(List of Strings)

-FA functionalities:

- a new FA object can be created by providing the filename of the file containing the necessary data

- void readSet(String line, List<String> set): splits the given line into tokens and adds the tokens to the given set

- void readTransition(String line): splits the given line into tokens and creates a new Transition with those tokens; adds the transition to the list of transitions

- void readFromFile(): reads FA from file having Filename given when the object was constructed

- boolean isDFA(): returns true if FA is deterministic and false otherwise

- String getNextState(String startState, String value): iterate through all transitions that have as startState of transition the given startState and as value of transition the given value and returns the endState or “false” if there was no match

- boolean isAccepted(String sequence): given a sequence checks if it is accepted by the FA; goes from one state to another by using getNextState function and finally checks if the lastState of the sequence is an endState from the set of end states of FA

- String toString(): overrides function toString 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>

