**Documentation**

For my Finite Automaton class I used:

* A string for the initialState;
* A list of strings in order to represent the states;
* A list of strings in order to represent the alphabet;
* A list of strings in order to represent the final states;
* A Map composed of Pair<String, String> and Set<String> in order to represent the transitions.

/\*\*

\* This method reads the content of the Finite Automaton from the file and populates the lists for the states, alphabet, finalStates, the string

\* for the initial state and the map for the transitions

\* @param filePath - the file path of the file which will be read

\*/

**private void readFromFile(String filePath)**

/\*\*

\* With this method we check if the FA is deterministic or not

\* @return true if the FA is deterministic, false otherwise

\*/

**public boolean checkIfDeterministic()**

/\*\*

\* @return the states of the FA

\*/

**public List<String> getStates()**

/\*\*

\* @return the initial state of the FA

\*/

**public String getInitialState()**

/\*\*

\* @return the alphabet of the FA

\*/

**public List<String> getAlphabet()**

/\*\*

\* @return the list of final states of the FA

\*/

**public List<String> getFinalStates()**

/\*\*

\* @return the transitions of the FA

\*/

**public Map<Pair<String, String>, Set<String>> getTransitions()**

/\*\*

\* With this method we check if a sequence is accepted by the finite automaton

\* @param sequence - the sequence we check if it's accepted

\* @return - true if the sequence is accepted and contained by the list of final states of the FA and false otherwise

\*/

**public boolean acceptsSequence(String sequence)**

**Lexic + Syntax for how we write the FA file:**

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

specialSymbol ::= '-'|'+'

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

alphabetCharacter ::= letter | digit | specialSymbol

alphabet ::= alphabetCharacter;{alphabetCharacter;}

state ::= letter

listOfStates ::= state;{state;}

transition ::= state alphabet state

listOfTransitions ::= transition'\n'{transition'\n'}

initialState ::= state

finalStates ::= state;{state;}

FAfile ::= listOfStates '\n' alphabet '\n' initialState '\n'finalStates '\' transition