https://github.com/mocadavid/LFTC/tree/Final-Lab4

Am copiat din nou documentatia de la laboratorul 4 si am adaugat descrierea pentru functia in plus. Documentatia pentru scanner nu se shimba la functii deloc, ci doar scannerul are 2 field-uri in plus care sunt cele 2 automate.

```
/**
* Class which works as a finite automata.
*/
public class FA {

//This field is true if the given sequence is accepted by the FA.
private boolean valid = false;
//The list of states.
private List<String> states = new LinkedList<>();
//The list of alphabet.
private List<String> alphabet = new LinkedList<>();
//The initial state.
private String initialState;
//The list of final states.
private List<String> finalStates = new LinkedList<>();
//The dictionary of transitions.
private Map<Pair,String> transitions = new HashMap<>();

  /**
   *Loads the states, alphabet, initial state, final states, transitions from a file.
   * @param filename: The name of the file: String
   */
  public void loadFile(String filename);
  /**
   *Main function of the FA.
   * @param filename:The file’s name on which the FA is built: String
   */
  public void start(String filename);

  /**
   *Carries UI logic on displaying the option that the user wants.
   */
  private void menu()

/**
*Checks if the sequence is accepted i.e. it is neither non-deterministic nor invalid for
the given fa.
* @param sequence The sequence we check: String
* @return the result of the check: true if it is deterministic and accepted by fa,
false otherwise
*/
public boolean isAccepted(String sequence);

  /**
   *Displays the options available for the user.
   */
  private void displayMenu();
}

//Class used in creating transitions
public class Pair {
  //State from which we go
  public String source;
  //State we reach
  public String destination;

  //constructor
  public Pair(String source, String destination);

  /**
   * Constructs a display string with all the arguments for the pair class.
   * @return Concatenation of source and destination: String
   */
  @Override
  public String toString();
}
```

EBNF of FA.in

```
fa::= states alphabetList initialState finalStates transitions
states::={state”,”}state
state::=”q” number
number::=”1”|...|”9”{“0”|....|”9”}
alphabetList ::={alphabet”,”}alphabet
```

alphabet::=”a”|...|”z”

initialState::=state

finalStates::={state”,”}state

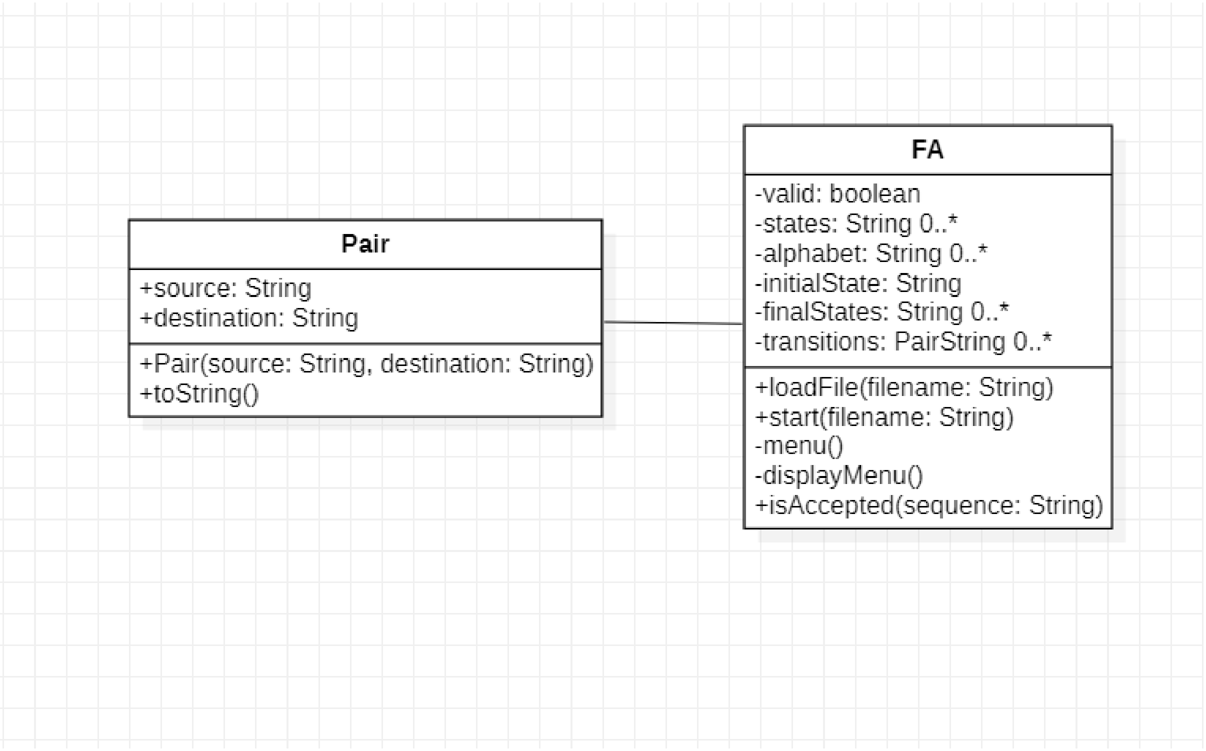
transitions::={rowTransition”\n”}rowTransition (nr rowTransition = nr states)

rowTransitions::={transition”,”}transition   (nr of transition = alphabetList size)

transition::=noTransition|state{multipleStates}

noTransition::=”-”

multipleStates::=”:”state


Pair
+source: String
+destination: String
+Pair(source: String, destination: String)
+toString()
FA
-valid: boolean
-states: String 0..*
-alphabet: String 0..*
-initialState: String
-finalStates: String 0..*
-transitions: PairString 0..*
+loadFile(filename: String)
+start(filename: String)
-menu()
-displayMenu()
+isAccepted(sequence: String)