Github Link: [https://github.com/916RasnitaRadu/SEM5-FLCD/tree/main/Lab4](https://github.com/916RasnitaRadu/SEM5-FLCD/tree/main/Lab3)

Documentation

Finite Automata

The FiniteAutomata class implements a finite automation and its operations read from a file. It is composed of:

* states: List<String> = the list of states of the FA;
* alphabet: List<String> = the alphabet of the FA;
* outputStates: List<String> = the states of the FA that are marked as finals;
* transitions: List<Transition> = a list of 3-tuples in which first is the “from”- state, second is the “to”-state, and the third is the label of the transition;
* initialState: String = initial state of the FA
* filename: String = the name of the given file, passed in constructor;
* labelsRegex: Pattern = a regex for parsing the file

Operations

* init( ) : void – reads the given file and constructs the states, alphabet, outputStates, initialState, transitions; throws an Exception if the file is not correct written;
* printTransitions( ) : void – display the list of transitions
* printInitialState( ) : void – display the initial state
* printOutputStates( ) : void – display the final states
* printAlphabet( ) : void – displays the alphabet
* printStates( ) : void – display the list of states
* checkSequence(word : String) : Boolean – verifies if the given sequence (sequence of alphabet elements) is accepted by the finite automata (i.e. starting from the initial state, we reach the final state)
* getNextAcceptedSequence(word : String) : String – get the substring of the input sequence that is accepted by the FA (starting from the beginning of it)

Transition

This class represents a transition used in the FiniteAutomata class. It has 3 fields of type String: “from”, “to” and “label” (these were explained above) and getters for these fields. Basically a Transition object is a 3-tuple of form (from, to, label).

EBNF

letter = "a"|"b"|...|"z"|"A"|"B"|...|"Z"

digit = "0"|"1"|...|"9"

non\_zero\_digit = "1" | "2" | ... | "9"

integer = "0" | ["+" | "-"] non\_zero\_digit {digit}

unsigned\_integer = non\_zero\_digit {digit}

char = "'"letter"'" | "'"digit"'"

string = """ {char} """

firstLine = "states" "=" "{" {char} {"," char} "}"

secondLine = "initial\_state" "=" {char}

thirdLine = "out\_states" "=" "{" {char} {"," char} "}"

fourthLine = "alphabet" "=" "{" {char} {"," char} "}"

triple = “(“ {char} “,” {char} “,” {char} “)”

fifthLine = "transitions" "=" "{" {triple} {"," triple} "}"

file = firstLine “\n” secondLine “\n” thirdLine “\n” fourthLine “\n” fifthLine