Lab 4 - Documentation

https://github.com/cs-ubbcluj-ro/lab-work-computer-science-2024-georgianapetricele/upload/main/1-Mini-Language-And-Scanner/Lab4

Finite Automata

The Finite Automata is structured as a class with 5 fields:

- „states" – for Q
- „alphabet"- for E
- „initial state" – for q0
- „final states" – for F
- „transitions" – for T

Operations:

public void readFromFile() throws IOException

- Reads the finite automata's definition (states, alphabet, initial state, final states, and transitions) from a file. It also constructs the list of transitions

private void printListOfString(String listname, List<String> list)

- Prints a list of strings, used for printing the states, the alphabet or final states.

public boolean isDFA()

- Checks if the finite automaton is a Deterministic Finite Automata (DFA). It returns false if there are multiple transitions for the same state-symbol pair; otherwise, it returns true.

public boolean checkAcceptedSequence(String word)

- Checks if the given input word is accepted by the DFA. It starts from the initial state, processes each character of the word according to the transitions, and returns true if it ends in a final state; otherwise, false

public String getNextAccepted(String word)

- Processes the given input word and returns the longest prefix that can be accepted by the automaton. If the prefix ends in a final state, it returns the accepted part of the word; otherwise, it returns null.

Transition

This class is used for representing a transition. It has 3 fields: from, to and label. It basically works like a weighted graph and you are going from node 1 to node 2 with a given weight. The transitions have the form(from, to, label)

EBNF

non_zero_digit = 1|2|..|9

digit = 0|1|…|9

number = non_zero_digit{digit}

letter = a|b|…|z|A|B|…|Z

character = letter|digit

firstLine =“states”“:”{character}{“,” character}

secondLine=”alphabet”“:”{character}{“,”character}

thirdLine = “initial state” “:” {character}

fourthLine=“final states” “:” {character}{“,”character}

transition = {character} “ “ {character} “ ”{character}

fifthLine = “transitions” “:” {transition} {“,” transition}

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