GitHub Link -Geanovu Medeea-Elena 933/1

Finite Automata Implementation

The class FA adapt the Finite Automata model, having as variables a set for the states, a set fot the alphabet, a map for the productions (firststate + symbol -> resultstate), the initial state and a set of final states.

-addTransition(fromState, symbol, toState): void – adds a new transition

-isDFA(): bool – checks each transition for any doubles, which would cause non-deterministic behaviour of the method

-acceptsSequence(sequence): bool – checks if each character of the sequence has a transition. It starts from the initial state, and computes the productions with each character of the sequence consecutively, and in the end it checks wheter or not the productions end in a final state.

EBNF form of the FA.in contents:

letter = ”A” | ”B” | … | ”Z” | … |”a” | ”b” | … | ”z”

digit = ”0” | ”1” | … | ”9”

special_character = ”+” | ”-” | ”.” | ”_”

finite_automaton = states, alphabet, transitions, initial_state, final_states

states = ”[states]” state {”,” state}

state = ”q” digit {digit}

alphabet = ”[alphabet]” symbol {”,” symbol}

symbol = letter | digit | special_character

transitions = ”[transitions]” transition {transtition} ”end”

transition = state ”,” symbol ”,” state

initial_state = ”[intialState]” state

final_states = ”[finalStates]” state {”,” state}