Student: Condrea Adrian

Lab 4 - Finite Automata
Documentation

Github link: https://github.com/adriancondrea/Lab4_FiniteAutomata

Program functionalities:

1. Reads the elements of a FA (from file)
2. Displays the elements of a finite automata, using a menu: the set of states, the alphabet, all the transitions, the set of final states, the initial state.
3. For a DFA, verify if a sequence is accepted by the FA.

Implementation details:

Finite Automata class consists of a state, a boolean function canStop which returns whether the state is final or not, and a switch state function which takes as input an action and performs the transition of the current state by that action.

The state class contains a list of transitions, and a boolean field isFinal. PerformTransition function looks for a transition possible from current state, by performing given action. If such transition exists, we move to the next state. Otherwise, we throw an exception.

A transition consists of an action (how we get to the next state) and a new state. The isPossible method takes as parameter a sequence of characters and check if it is equal to the action field of the instance.

We read the Finite Automata from a text file and display the options menu:

1. Display the set of states
2. Display the alphabet
3. Display all the transitions
4. Display the set of final states
5. Verify if a sequence is accepted by the DFA
6. Display the initial state

When verifying if a sequence is accepted by the DFA, we check that we have parsed the entire stack, and that we are in a final state.

EBNF form for the input file (FA.in):

On the first line, we will have states, separated by “,”:

states = {state}
state = {character} | {character} “,” state
character = digit | letter
digit = “0” | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | “9"
letter = “a” | “b” | .. | “z” | “A” | “B” | … | “Z”

On the second line, we will have the alphabet items, separated by “,”:

alphabet = {alphabet_item}
alphabet_item = {character} | {character} “,” alphabet_item

On the third line, we will have the initial state:

initial_state = “q0 = “ state

On the fourth line, we will have the set of final states:

final_states = “F = “ {state}

On the fifth line, we will have the set of transitions:

transitions = “S = “ {transition}
transition = state “,” alphabet_item “->” state | state “,” alphabet_item “->” state “,” transition