

Documentation - lab4

Finite Automata

<https://github.com/iuliaaai/LFTC/tree/master/lab4>

Finite automata is a class containing fields for all the states(set of strings), initial state(string), final states(set of strings), the alphabet(set of strings), and the transitions. The transitions are stored as a map having the key as a pair of two strings(the source state and the value needed to access the destination state), each mapped to a set of strings, representing the destination states.

Checking if a FA is DFA is made by verifying that the sets of strings from the transition map have a length of 1(there is no state that can access more than 1 destination states with the same value).

For a DFA, verifying if a sequence is accepted by the FA is done as follows: we start from the initial state, go through each character of a given sequence and check if the pair formed by the current state and the current character in the sequence is mapped to a set containing only one value(otherwise the sequence is not accepted by the FA and we stop). This value will be the starting point for the next iteration. We go like this until we reach the end of the sequence and if the last character is a final state, it means the sequence is accepted by the FA.

digit ::= '0' | '1' | ... | '9'

letter ::= 'a' | 'b' | ... | 'z' | 'A' | ... | 'Z'

alphabetChar ::= letter | digit

state ::= letter

states ::= state {state}

alphabet ::= alphabetChar {alphabetChar}

transition ::= state alphabet state

transitions ::= transition {transition}

initialState ::= state

finalState ::= state {state}

FA ::= states \n alphabet \n initialState \n finalState \n transitions