# FiniteAutomaton Class Documentation

The FiniteAutomaton class represents a finite automaton (FA) that can be used to model and evaluate sequences based on a set of defined states, transitions, and an alphabet. The class allows the creation and manipulation of deterministic and non-deterministic finite automata, with the capability to check if a given input sequence is accepted by the automaton.

## Class Constructor

FiniteAutomaton()

Initializes the finite automaton with an empty set of states, alphabet, transitions, an initial state set to None, and an empty set of final states.

## Methods

### add\_state(state, is\_final=False)

Adds a new state to the automaton. If is\_final is True, the state is also added to the final states.

### add\_transition(from\_state, symbol, to\_state)

Defines a transition in the automaton from a given state using a specified symbol to a destination state. This helps in mapping how the automaton should progress through states based on input symbols.

### set\_initial\_state(state)

Sets the initial state of the automaton, which serves as the starting point for sequence evaluation.

### is\_accepted(input\_string)

Evaluates whether a given input\_string is accepted by the finite automaton. The method processes the input sequence symbol by symbol, following transitions to reach a final state if possible. If the ending state is an accepting state, the input string is valid for the automaton.

### check\_if\_deterministic()

Verifies if the automaton is deterministic. A deterministic finite automaton (DFA) has at most one unique transition for each symbol in every state. If any state has multiple transitions for a symbol, the automaton is considered non-deterministic.

### write\_transitions()

Returns a string representation of all transitions in the automaton in the format 'from\_state --symbol--> to\_state'. This can be used to visualize or log the transitions.

### display()

Displays a summary of the automaton's states, alphabet, initial state, final states, and transitions. This provides a quick overview of the automaton configuration.

### parse\_fa\_file(fa\_file)

Parses a finite automaton definition from a text file. The file should contain states, initial state, final states, and transitions, each line specifying a part of the automaton. This allows for easy setup of automata from predefined files.