Definition of the structure of the input file for the finite automaton:

<fa> ::= <initial_state> <transitions>

<initial_state> ::= '{' <state> '}' ',' <alphabet> '->' '(' <state> ')' | '[' <state> ']'

<transitions> ::= <transition> | <transition> <transitions>

<transition> ::= '(' <state> ',' <alphabet> ')' '->' '(' <state> ')' | '(' <state> ',' <alphabet> ')' '->' '[' <state> ']'

<state> ::= 'q' <number>

<alphabet> ::= <char> | <char> '-' <char>

<char> ::= 'a' | 'b' | 'c' | ... | 'z' | 'A' | 'B' | 'C' | ... | 'Z' | '0' | '1' | '2' | ... | '9'

<number> ::= '0' | '1' | '2' | ... | '9'

<fa> is the top-level rule and it represents the entire input file. It consists of an initial state and a list of transitions.

<initial_state> defines the rule for the initial state. It starts with an opening curly brace '{', followed by a state, a closing curly brace '}', a comma ',', and an alphabet. Then it ends with an arrow '->'.

<transitions> defines the rule for the list of transitions. It can consist of one or more transitions.

<transition> defines the rule for a single transition. It starts with an opening parenthesis '(', followed by a state, a comma ',', an alphabet, and a closing parenthesis ')'. Then it ends with an arrow '->' and either a final state or a non-final state.

<state> defines the rule for a state. It starts with the letter 'q' followed by a number.

<alphabet> defines the rule for the alphabet. It can be a single character or a range of characters separated by a dash '-'.

<char> defines the rule for a single character. It can be any uppercase or lowercase letter or any digit.

<number> defines the rule for a number. It can be any digit.