

## Documentation Lab 4

Github Link of FA.in: [https://github.com/timoteicopaciu/LFCD/blob/main/Lab\\_04/FA.in](https://github.com/timoteicopaciu/LFCD/blob/main/Lab_04/FA.in)

Github Link for code: [https://github.com/timoteicopaciu/LFCD/blob/main/Lab\\_04/lab\\_04.py](https://github.com/timoteicopaciu/LFCD/blob/main/Lab_04/lab_04.py)

```
class TableFA:
    """
    This class represent a FA as a table
    """
    def readFAFromFile(self, filename):
        """
        Read a FA from a file
        :param filename: string, the name of the name where FA is stored
        :preconditions: filename must to be a string, representing a file name
        :postconditions: the FA object's attributes will be completed
        :return: none
        """
    def print(self, x):
        """
        Print some attributes of FA
        :param x: char, representing an option in order to know what to return
        :preconditions: x must be a string, x is from A = {'1', '2', '3', '4'}
        :postconditions: a string is returned, representing an attribute as a string, or '' if x
        is not in A
        :return: a string
        """
    def isAccepted(self, sequence):
        """
        Verify if a sequence is accepted by the FA
        :param sequence: string, the sequence to be verified if is accepted by he FA
        :preconditions: sequence must to be a string
        :postconditions: return if the sequence is accepted or not
        :return: 'Is accepted!' or 'Is not accepted!' or 'Is not a DFA!'
        """
```

### EBNF of FA.in

fa ::= initialState finalStates alphabet states transitions

initialState ::= state

finalStates ::= {state,}state

alphabet ::= "A" | "B" | ... | "Z" | "a" | "b" | ... | "z" | "0" | "1" | ... | "9" | "-" | "\_"

states ::= {state,}state

state ::= "q" number

number ::= "1" | ... | "9" { "0" | "1" | ... | "9" }

transitions ::= transition { "\n" transition } – (nr of states \* len of alphabet) lines

transion ::= {state,}state