

# Finite Automatum

## Requirements:

Write a program that:

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.
3. For a DFA, verify if a sequence is accepted by the FA.

Link to github: <http://github.com/bogdan1809/FLCD>

FiniteAutomata
+ Q: list
+ E: list
+ q0: str
+ F: list
+ S: list
+ readLine: list
+ readFile
+ readTransition
+ isDeterministic: bool
+ isAccepted: bool
+ checkIfDeterministic: bool
+ printAll

1. The elements are read from the file line by line, in the following order:

- 1<sup>st</sup> line – Q : list of states
- 2<sup>nd</sup> line – E : list of elements
- 3<sup>rd</sup> line – q0: initial state
- 4<sup>th</sup> line – F: list of final states
- 5<sup>th</sup> line until end of file – S: list of transitions (each transition on a line)

```
def isDeterministic(self, transitions):
    lista = []
    transitionset = set()
    for transition in transitions:
        lista.append(transition[1])
        transitionset.add(transition[0])
    if len(set(transitionset)) == len(lista):
        return True

    return False
```

The `isDeterministic` function is taking the list of all transitions, making a set with the pair (state,symbol) and then checks that the length of this set is equal to the length of resulting states. If it's not, it means that a pair repeats itself -> not deterministic.

```
def isAccepted(self, sequence):
    if self.isDeterministic(self.S):
        state = self.q0

        while (len(sequence) > 0):
            elem = sequence[0]
            key = (state, elem)
            ok = 0
            for transition in self.S:
                if key == transition[0]:
                    state = transition[1]
                    sequence = sequence[1:]
                    ok = 1
            if ok == 0:
                return False
        if state in self.F:
            return True
        return False
```

The `isAccepted` function is taking a sequence as parameter. It first checks that the FA is deterministic, then changes its state to the initial state. It then takes the sequence element by element and checks if there is a transition that can be made. If the sequence finished and the current state is one of the final states, it returns True.

Examples:

```
Q = { q1, q2, q3 }
E = { 0, 1 }
q0 = q1
F = { q1, q3 }
S = {
    (q1, 0) -> q1,
    (q1, 1) -> q3,
    (q2, 0) -> q2,
    (q2, 1) -> q1,
    (q3, 0) -> q3,
```

```
(q3, 1) -> q2  
}
```

This FA is deterministic since there is no pair of (state,symbol) that can transition in more than one state.

Sequences that will be accepted: "0", "111"

Sequences that won't be accepted: "11", "1100"