

# Finite Automaton

Student: Alexandrescu Andrei-Robert, 931/1

Current lab: [Repository Link](#)

Integration with previous lab: [Repository Link](#)

## 1. File Structure (FA.in)

a) Mathematic (Natural Language) description

$s_1 s_2 \dots s_n$  (states)

$a_1 a_2 \dots a_m$  (alphabet)

noTran (number of transitions)

$p_1 b_1 q_1$  ( $\delta(p_1, b_1) = q_1$ )

...

$p_{noTran} b_{noTran} q_{noTran}$

$q_0$  (initial state)

$f_1 f_2 \dots f_o$  (final states)

b) EBNF:

identifier ::= letter | letter { letter | digit }

letter ::= "A" | "B" | . . . | "Z"

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

non\_zero\_digit ::= "1" | ... | "9"

constno ::= [ ("+" | "-") non\_zero\_number | zero

zero ::= 0

non\_zero\_number ::= non\_zero\_digit { digit }

states ::= identifier | identifier { identifier }

alphabet ::= { constno }

transitions ::= { identifier constno identifier }

initialState ::= identifier

finalStates ::= identifier { identifier }

c) Example:

p q r

a b

4

p a q

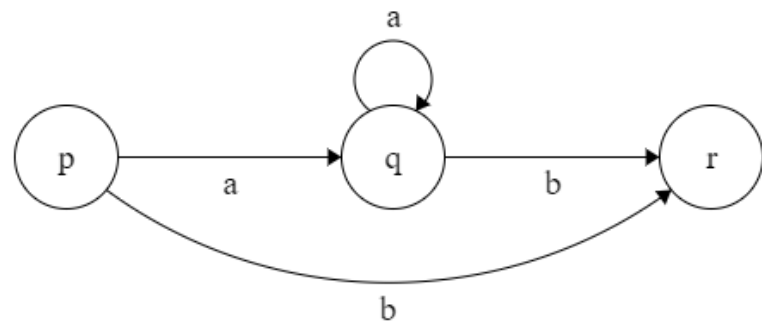
q a q

q b r

p b r

p

r



## 2. Program details

Method *readFA* is used to read the data from the *FA.in* file and store it accordingly in the RAM. Some error cases are treated such as:

- one of the transition terms (state 1, transition term, state 2) does not belong to the declared states / alphabet respectively
- the initial state does not belong to the declared states
- one of the final states does not belong to the declared states

Method *verifySequence* checks whether a given sequence is accepted by the FA. This is done by simply using a for loop to cycle through the characters of the sequence and using a *currentState* variable to keep track of the current state. The method *move* is used to transition between states using the current symbol from the alphabet. In case the sequence could not be consumed entirely, an error occurs.

## 3. Integration with lab 1-3

todo