Github link: https://github.com/stefa13/FLCD/tree/master/Lab4

I represented the finite automaton as a class that contains 5 attributes:

- the set of states (list of strings)

- the set of symbols from the alphabet (list of strings)

- the initial state (string)

- the set of final states (list of strings)

- the set of transitions, which maps a pair of type (source, route) to a list of destinations

The FA class has a method that reads a FA from file, which returns the FA created from the input file if it is valid, or throws an exception otherwise.

I also implemented in the FA class the following methods:

- a method that checks if the FA is a deterministic, which checks that for all destination states, there is no more than one route that reaches it.

- a method that checks if a sequence is accepted or not by the FA. This is done in the following way: I take the initial state of the FA as the current state and then iterate through the characters of the given sequence. If the pair formed of the current state and the current character from the sequence is a valid transition, then I make the current state to be the current character, otherwise, if it is not a valid transition, it means that the sequence is not accepted and I return false. If the last current state is also a valid transition, it means that the sequence is accepted by the FA and the method returns true.

The structure of the file is the following:

- first line: the set of states, separated by space

- second line: the set of symbols from the alphabet, separated by space

- third line: the initial state

- fourth line: the set of final states, separated by space

- all the remaining lines will contain a transition of the form: source, route, destination

The BNF is the following:

fa = states ‘\n’ alphabet ‘\n’ initial\_state ‘\n’ final\_states ‘\n’ transitions

states = state | state ‘ ‘ states

alphabet = symbol | symbol ‘ ‘ alphabet

initial\_state = state

final\_states = state | state ‘ ‘ final\_states

transitions = transition | transition ‘\n’ transitions

transition = state ‘, ’ symbol ‘, ’ state

character = letter | digit

string = {character}

letter = uppercase\_letter | lowercase\_letter

uppercase\_letter = "A" | "B" | ... | "Z"

lowercase\_letter = "a" | "b" | ... | "z"

digit = “0” | "1" | ... | "9"

state = string

symbol = string

Example of FA.in file:

q a b k

0 1

q

q a k

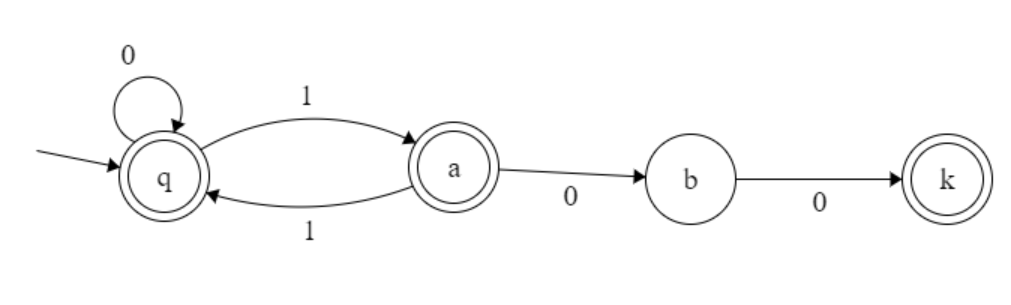
q, 0, q

q, 1, a

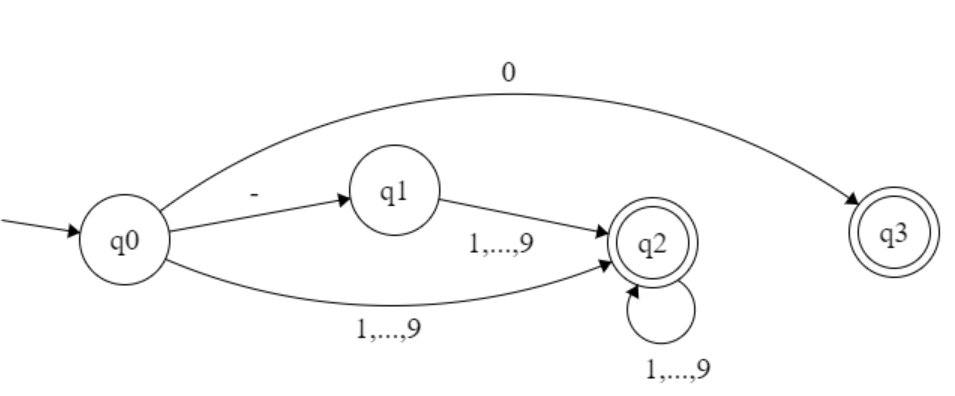
a, 1, q

a, 0, b

b, 0, k



FA for integer constants:



FA for identifiers:

