Práctica 2

Pedro Antonio Aguilar Lima

2022-2023

1 Ejercicios

1. Consider the language over the alphabet {a, b} that only contains the string a.

a) Build a DFA that recognizes this language and rejects all those strings that do not belong to the language.

DFA(Autómata finito determinista), está compuesto por:

(K,$\sum$,δ,s,F)
c K-> Conjunto de estados vacíos

$\sum$-> Alfabeto

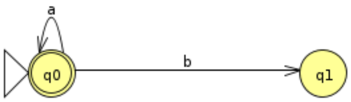
$s \in K$-> Estado inicial

$F \subseteq K$-> Conjunto de estados finales

$\delta : K \times \sum \rightarrow K$-> Función de transición

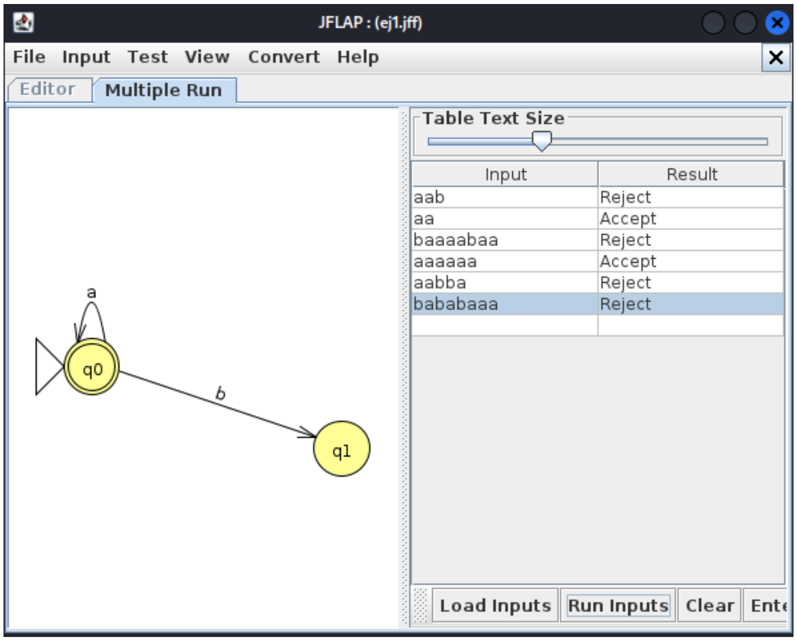
Un ejemplo reconocedor del lenguaje puede ser el siguiente:

M=({q_0,q_1},{a,b},δ,q_0,{q_0})



b) Test the automaton that you have created by introducing 6 chains.

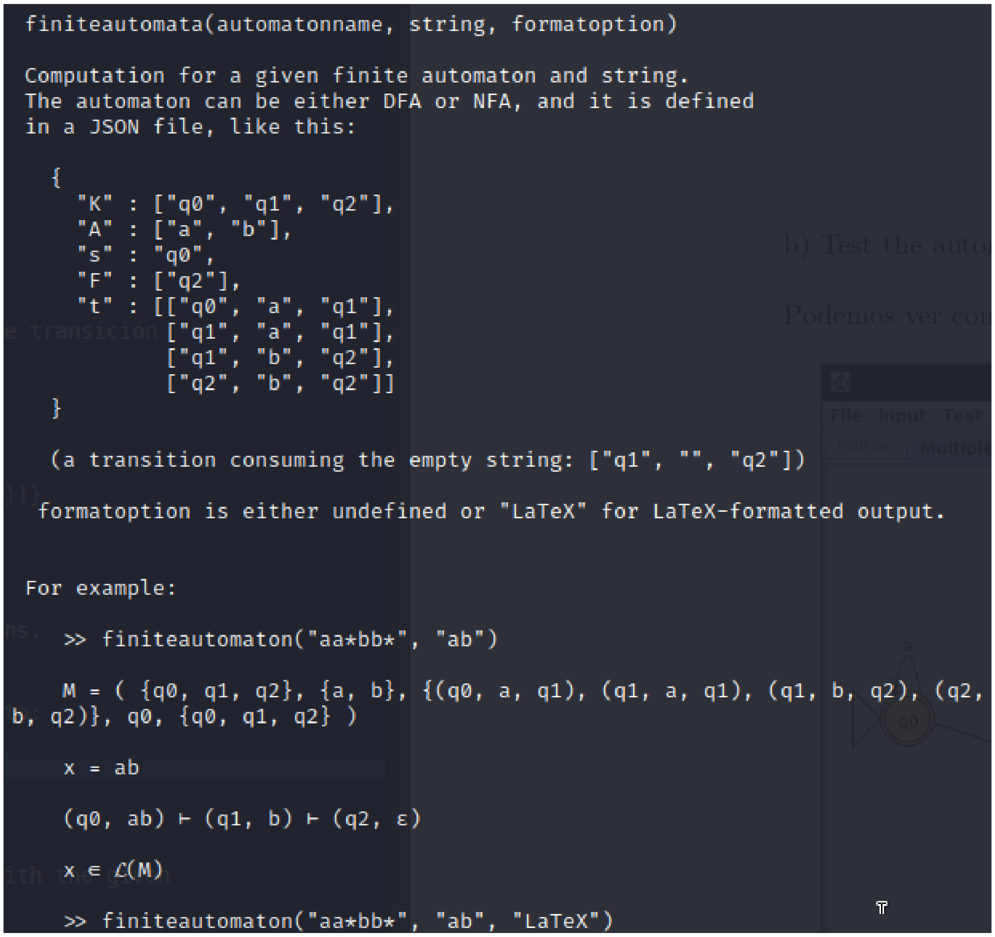
Podemos ver como acepta o rechaza las cadenas introducidas de texto:



Input	Result
aab	Reject
aa	Accept
baaaabaa	Reject
aaaaaa	Accept
aabba	Reject
bababaaa	Reject

2. Finite automaton in Octave:

a) Open the Octave **finiteautomata.m** script and test it with the given example (see script help) in the GitHub repository.



```
finiteautomata(automatonname, string, formatoption)

Computation for a given finite automaton and string.
The automaton can be either DFA or NFA, and it is defined
in a JSON file, like this:

  {
    "K" : ["q0", "q1", "q2"],
    "A" : ["a", "b"],
    "s" : "q0",
    "F" : ["q2"],
    "t" : [["q0", "a", "q1"],
           ["q1", "a", "q1"],
           ["q1", "b", "q2"],
           ["q2", "b", "q2"]]
  }

  (a transition consuming the empty string: ["q1", "", "q2"])

 formatoption is either undefined or "LaTeX" for LaTeX-formatted output.


For example:

   >> finiteautomaton("aa*bb*", "ab")

   M = ( {q0, q1, q2}, {a, b}, {(q0, a, q1), (q1, a, q1), (q1, b, q2), (q2,
b, q2)}, q0, {q0, q1, q2} )

   x = ab

   (q0, ab) ⊢ (q1, b) ⊢ (q2, ε)

   x ∈ ℒ(M)

   >> finiteautomaton("aa*bb*", "ab", "LaTeX")
```

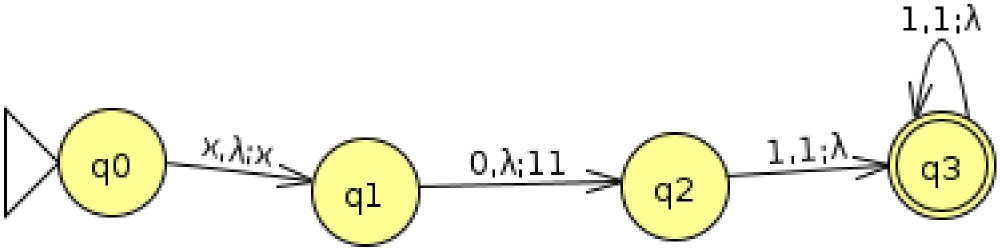
b) Specify in **finiteautomata.json** the automaton created in Activity 1 and test it with the script!

```
{
”K” : [”q0”, ”q1”, ”q2”],
”A” : [”a”, ”b”],
”s” : ”q0”,
”F” : [”q1”],
”t” :
[[”q0”, ”a”, ”q1”],
[”q0”, ”b”, ”q2”],
[”q1”, ”a”, ”q2”],
[”q1”, ”b”, ”q2”],
[”q2”, ”a”, ”q2”],
[”q2”, ”b”, ”q2”]]
}
```

3. Test the Free Context Pumping Condition for the first three examples.

Al no ser lenguajes libres de contexto, vamos a ver que no tenemos una partición de w.

4. Build an NPDA that recognizes the language $L = \{0^n 1^{2n} : \text{n} > 0\}$.


q0
x,λ;x
q1
0,λ;11
q2
1,1;λ
q3
1,1;λ