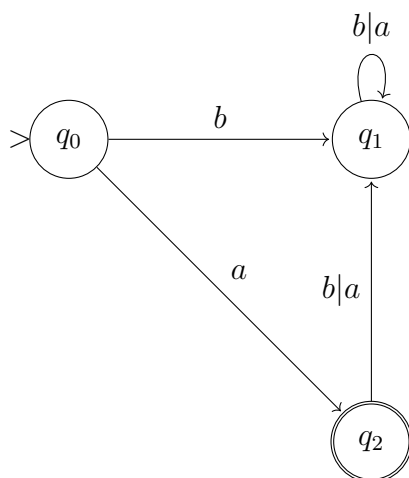


Ejercicio 1:

a) Sea $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

$\delta(q, \sigma)$	a	b
q_0	q_2	q_1
q_1	q_1	q_1
q_2	q_1	q_1

Por tanto el dirigrafo que representa al autómata es:



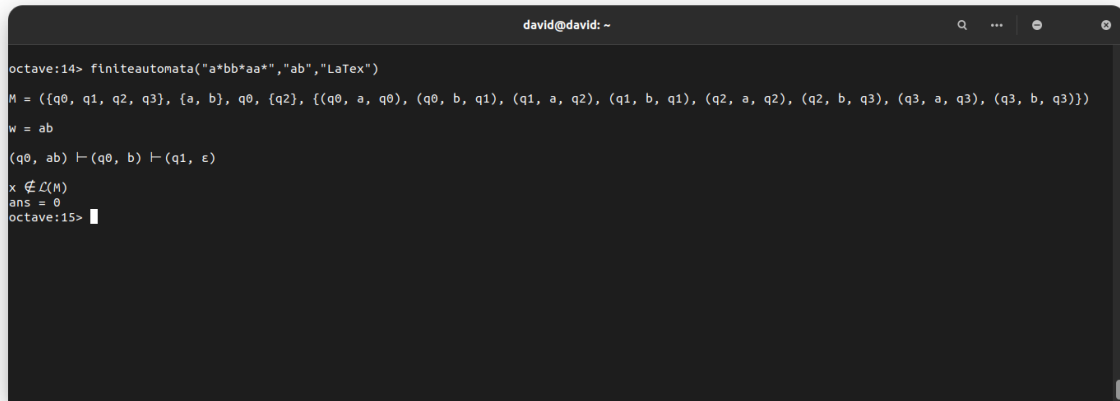
b) Con el programa JFLAP comprobamos el grafo descrito con 6 cadenas:

The screenshot shows the JFLAP interface with the automaton diagram on the left and a table of test results on the right.

Input	Result
aababba	Reject
a	Accept
aa	Reject
ab	Reject
b	Reject
ba	Reject

Ejercicio 2:

a) Probamos la función definida en el repositorio:



```

david@david: ~
octave:14> finiteautomata("a*bb*aa*", "ab", "LaTeX")
M = ({q0, q1, q2, q3}, {a, b}, q0, {q2}, {(q0, a, q0), (q0, b, q1), (q1, a, q2), (q1, b, q1), (q2, a, q2), (q2, b, q3), (q3, a, q3), (q3, b, q3)})
w = ab
(q0, ab) ⊢ (q0, b) ⊢ (q1, ε)
x ∉ L(M)
ans = 0
octave:15>

```

Una vez ya sabemos como funciona, definimos el autómata desde el archivo llamado "finiteautomata.json" tal como:

```

[
{
  "name" : "a*bb*aa*",
  "representation" : {
    "K" : ["q0", "q1", "q2", "q3"],
    "A" : ["a", "b"],
    "s" : "q0",
    "F" : ["q2"],
    "t" : [
      ["q0", "a", "q0"],
      ["q0", "b", "q1"],
      ["q1", "a", "q2"],
      ["q1", "b", "q1"],
      ["q2", "a", "q2"],
      ["q2", "b", "q3"],
      ["q3", "a", "q3"],
      ["q3", "b", "q3"]
    ]
  }
},
{
  "name" : "aa*bb*",
  "representation" : {
    "K" : ["q0", "q1", "q2"],
    "A" : ["a", "b"],
    "s" : "q0",
    "F" : ["q2"],
    "t" : [
      ["q0", "a", "q1"],
      ["q1", "a", "q1"],

```

```

        ["q1", "b", "q2"],
        ["q2", "b", "q2"]
    }
},
{
    "name" : "even_number_of_ones",
    "representation" : {
        "K" : ["q0", "q1"],
        "A" : ["0", "1"],
        "s" : "q0",
        "F" : ["q0"],
        "t" : [
            ["q0", "0", "q0"],
            ["q0", "1", "q1"],
            ["q1", "0", "q1"],
            ["q1", "1", "q0"]
        ]
    }
},
{
    "name" : "odd_number_of_zeroes",
    "representation" : {
        "K" : ["q0", "q1"],
        "A" : ["0", "1"],
        "s" : "q1",
        "F" : ["q0"],
        "t" : [
            ["q0", "0", "q1"],
            ["q0", "1", "q0"],
            ["q1", "0", "q0"],
            ["q1", "1", "q1"]
        ]
    }
},
{
    "name" : "lenguaje_con_a",
    "representation" : {
        "K" : ["q0", "q1", "q2"],
        "A" : ["a", "b"],
        "s" : "q0",
        "F" : ["q2"],
        "t" : [
            ["q0", "a", "q2"],
            ["q0", "b", "q1"],
            ["q1", "a", "q1"],
            ["q1", "b", "q1"],
            ["q2", "a", "q1"],
            ["q2", "b", "q1"]
        ]
    }
}
]

```

Una vez ya definido, ya lo podemos utilizar en nuestro programa Octave con la función ya definida "finiteautomata()":

```
david@david: ~
octave:18> finiteautomata("lenguaje_con_a","ab","LaTeX")
M = ([q0, q1, q2], {a, b}, q0, {q2}, {(q0, a, q2), (q0, b, q1), (q1, a, q1), (q1, b, q1), (q2, a, q1), (q2, b, q1)})
w = ab
(q0, ab) ⊢ (q2, b) ⊢ (q1, ε)
x ∉ L(M)
ans = 0
octave:19> finiteautomata("lenguaje_con_a","a")
M = ([q0, q1, q2], {a, b}, q0, {q2}, {(q0, a, q2), (q0, b, q1), (q1, a, q1), (q1, b, q1), (q2, a, q1), (q2, b, q1)})
w = a
(q0, a) ⊢ (q2, ε)
x ∈ L(M)
ans = 1
octave:20> finiteautomata("lenguaje_con_a","bbaaba")
M = ([q0, q1, q2], {a, b}, q0, {q2}, {(q0, a, q2), (q0, b, q1), (q1, a, q1), (q1, b, q1), (q2, a, q1), (q2, b, q1)})
w = bbaaba
(q0, bbaaba) ⊢ (q1, baaba) ⊢ (q1, aaba) ⊢ (q1, aba) ⊢ (q1, ba) ⊢ (q1, a) ⊢ (q1, ε)
x ∉ L(M)
ans = 0
octave:21> finiteautomata("lenguaje_con_a","aaaaaaaa")
M = ([q0, q1, q2], {a, b}, q0, {q2}, {(q0, a, q2), (q0, b, q1), (q1, a, q1), (q1, b, q1), (q2, a, q1), (q2, b, q1)})
w = aaaaaaaaa
(q0, aaaaaaaaa) ⊢ (q2, aaaaaa) ⊢ (q1, aaaaa) ⊢ (q1, aaaa) ⊢ (q1, aaaa) ⊢ (q1, aa) ⊢ (q1, a) ⊢ (q1, ε)
x ∉ L(M)
ans = 0
octave:22> finiteautomata("lenguaje_con_a","bbbbba")
M = ([q0, q1, q2], {a, b}, q0, {q2}, {(q0, a, q2), (q0, b, q1), (q1, a, q1), (q1, b, q1), (q2, a, q1), (q2, b, q1)})
w = bbbbbba
(q0, bbbbbba) ⊢ (q1, bbbba) ⊢ (q1, bbba) ⊢ (q1, bba) ⊢ (q1, ba) ⊢ (q1, a) ⊢ (q1, ε)
x ∉ L(M)
ans = 0
octave:23> █
```

Como vemos, lo he probado con 5 cadenas y la función nos dice que el autómata solo acepta, entre las cadenas introducidas, la cadena compuesta por una única a.