

Macchine di Turing

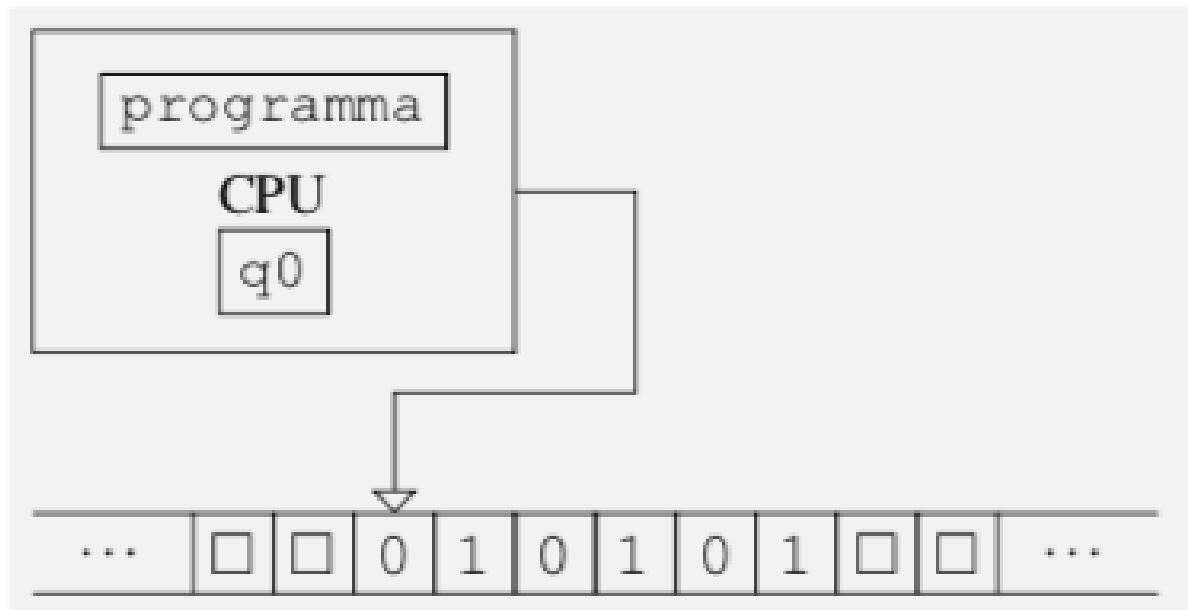
$$M = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$$

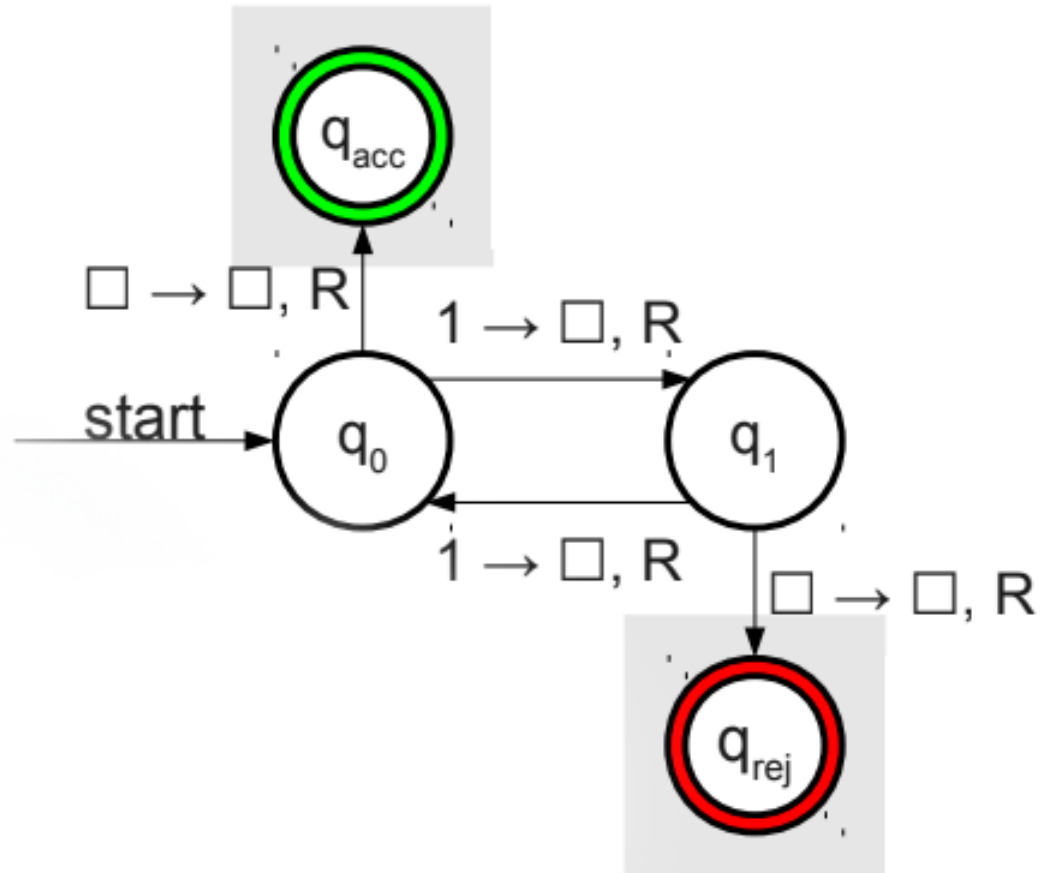
- $Q$ : insieme finito e non vuoto di stati
- $\Sigma$ : alfabeto di input
- $\Gamma$ : alfabeto dei simboli di nastro
- $\delta$ : funzione di transizione
- $q_0$ : stato iniziale
- $\square$ : simbolo di blank
- $F$ : insieme degli stati finali

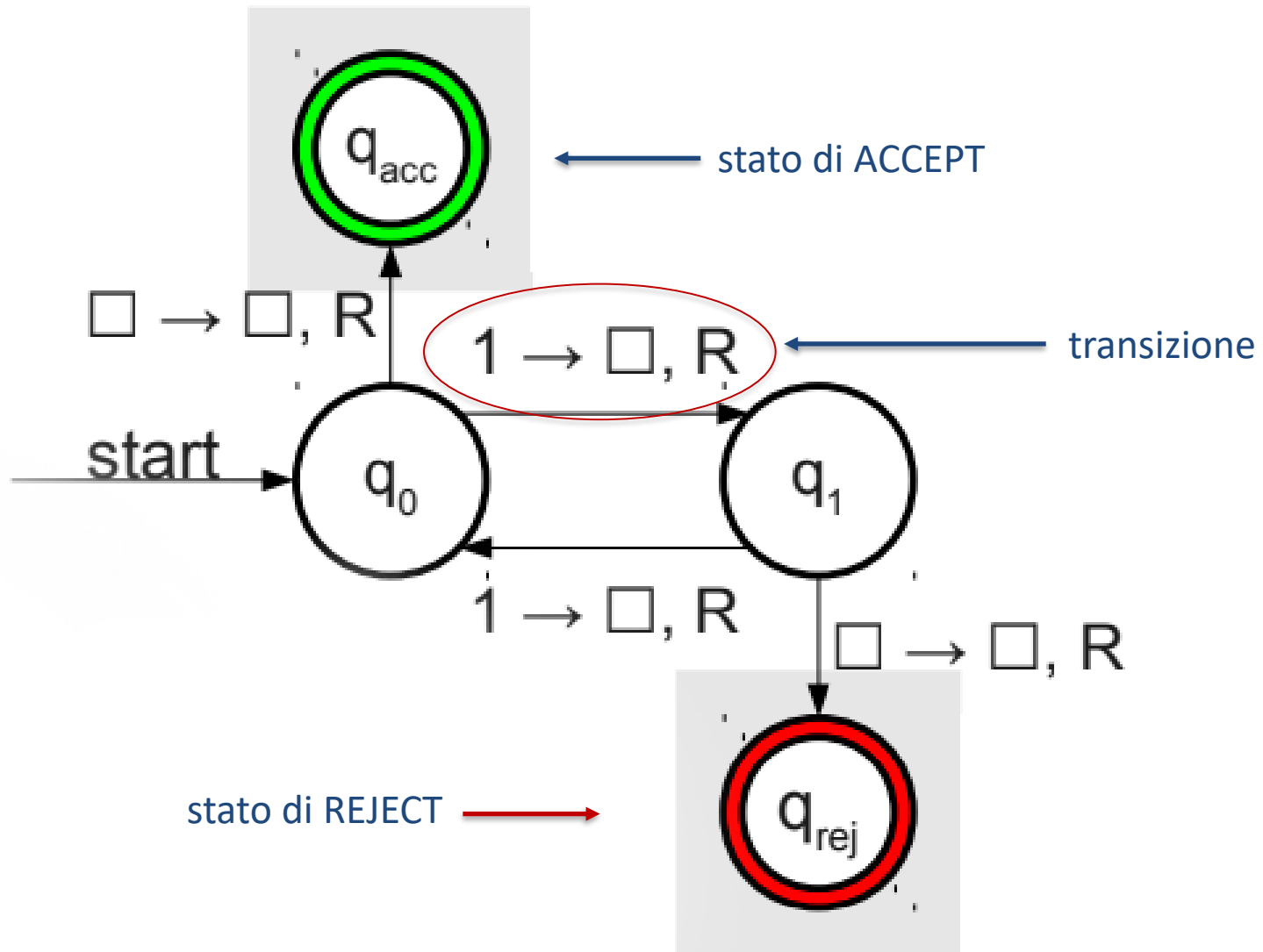
La CPU di una mdT è composta da:

- un registro di stato
- un programma con le istruzioni.

La memoria di una mdT è composta da un nastro infinito, diviso in celle.





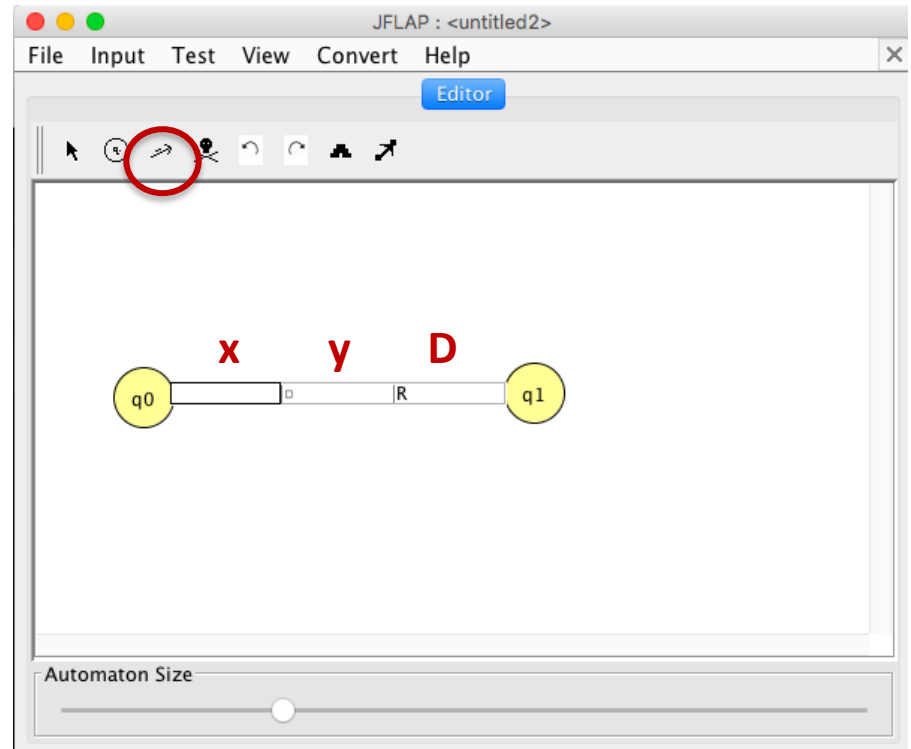
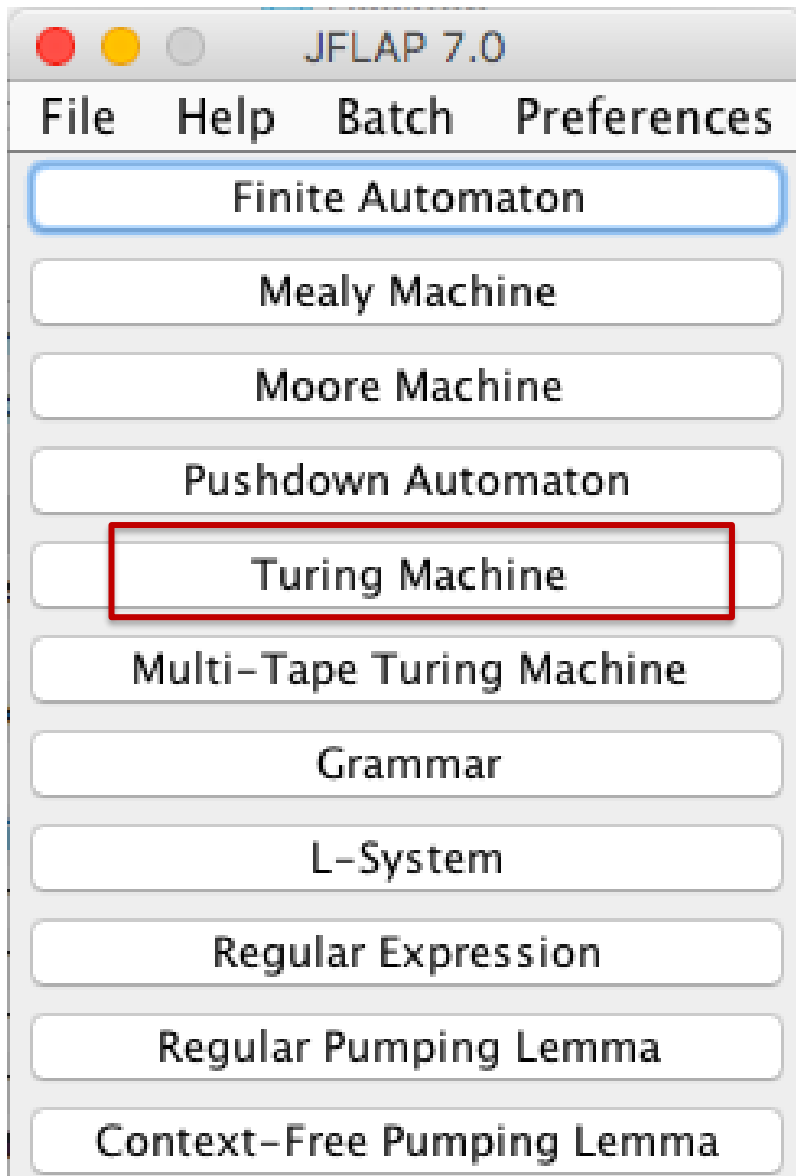


Transizione:  $x \rightarrow y, D$

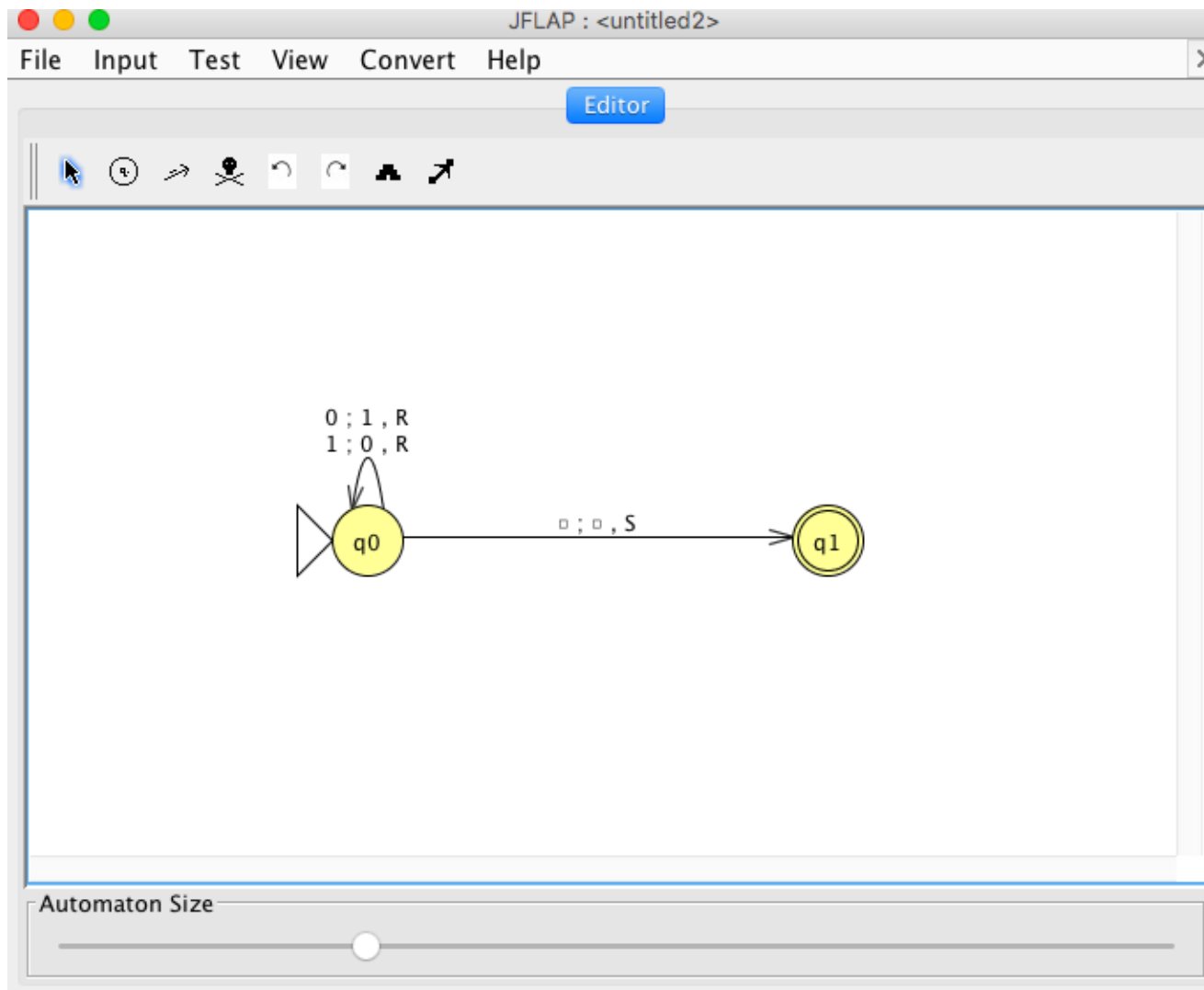
Alla lettura di  
(Read)

Sostituire con  
(Write)

Direzione  
(Left (L), Right (R), Stay (S))



Esercizio 1: costruire una mdT per calcolare il complemento a 1 di un numero binario (sostituire 0 con 1 e 1 con 0).





JFLAP : <untitled10>

File **Input** Test View Convert Help

- Step... ⌘R
- Step by BuildingBlock ⌘B
- Fast Run...
- Multiple Run ⌘M**
- Multiple Run (Transducer) ⌘T

Automaton Size

JFLAP : <untitled10>

Test View Convert Help

Editor **Multiple Run**

Table Text Size

Input	Result
01	Accept
11	Accept
aa	Reject

Load Inputs Run Inputs Clear Enter Lambda **View Trace**

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
01	Accept
11	Accept
aa	Reject

Traceback

↓

q1

oooooooooooooooooooo10oooooooooooooooooooo

1 : 0, R  
0 : 1, R

q0 → q1

∅; ∅, S

Load Inputs Run Inputs Clear Enter

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
01	Accept
11	Accept
aa	Reject

Traceback

↓

q1

oooooooooooooooooooo00oooooooooooooooooooo

1 : 0, R  
0 : 1, R

q0 → q1

∅; ∅, S

Load Inputs Run Inputs Clear Enter Lambda View Trace

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

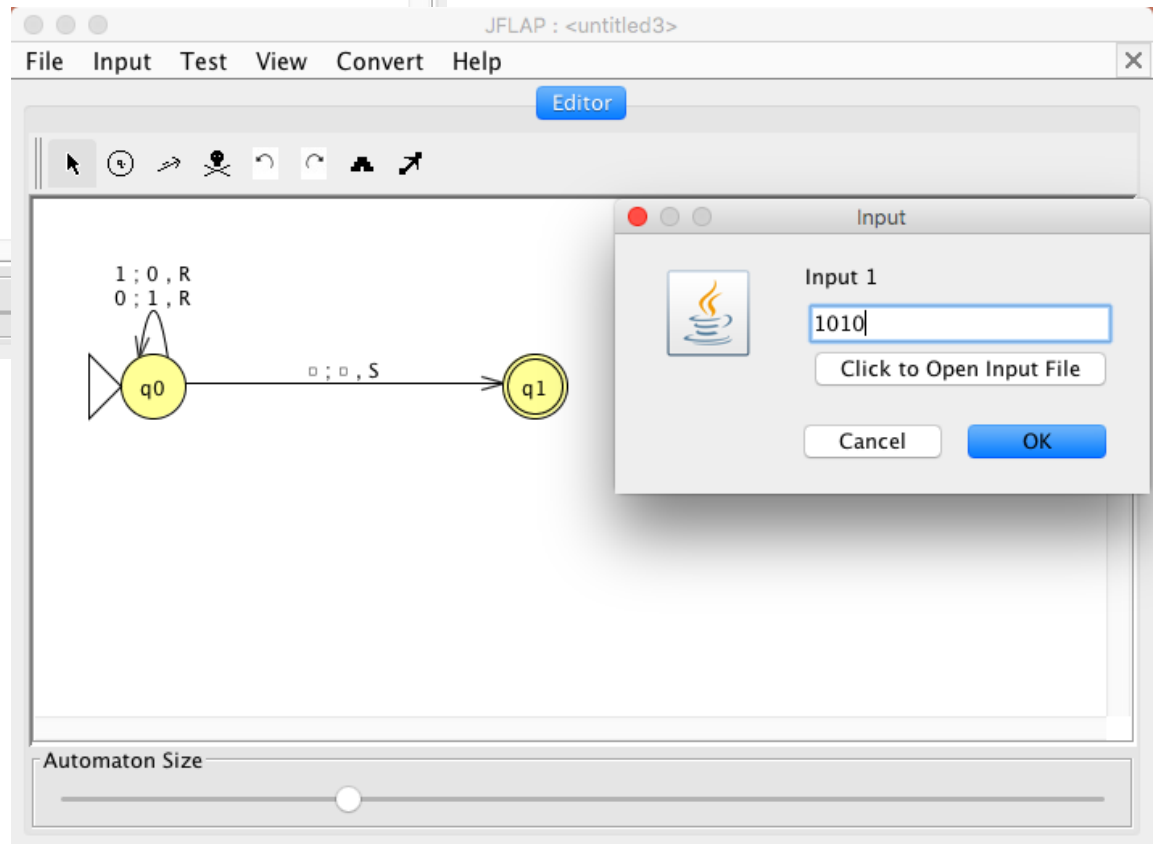
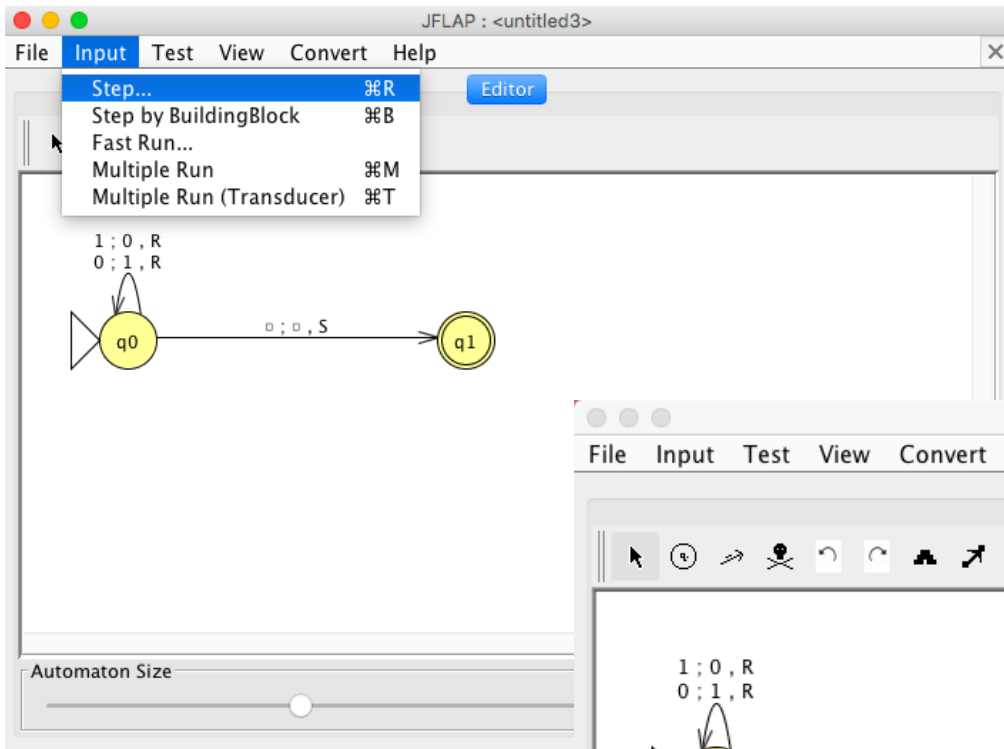
Input	Result
01	Accept
11	Accept
aa	Reject

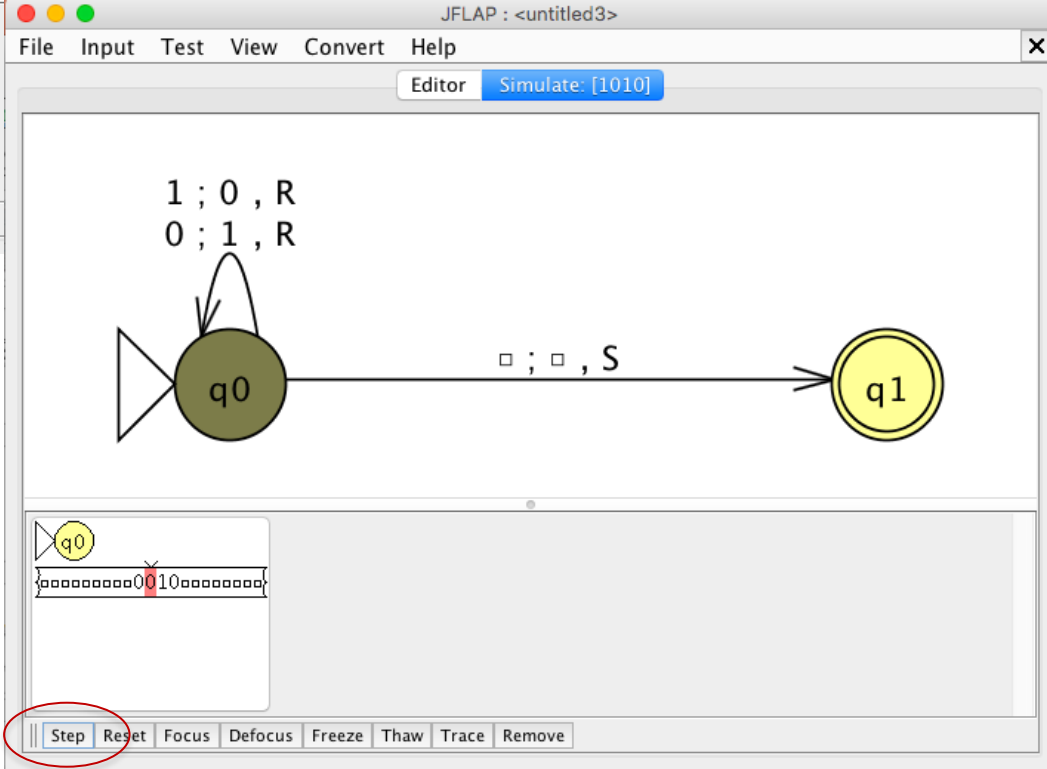
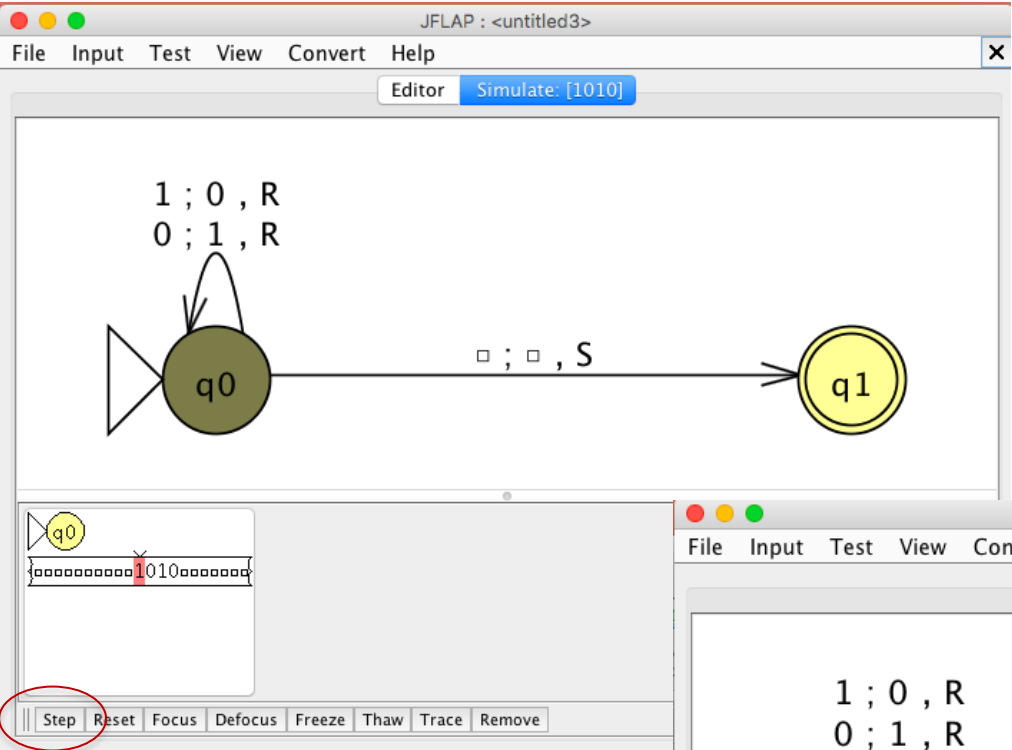
Traceback

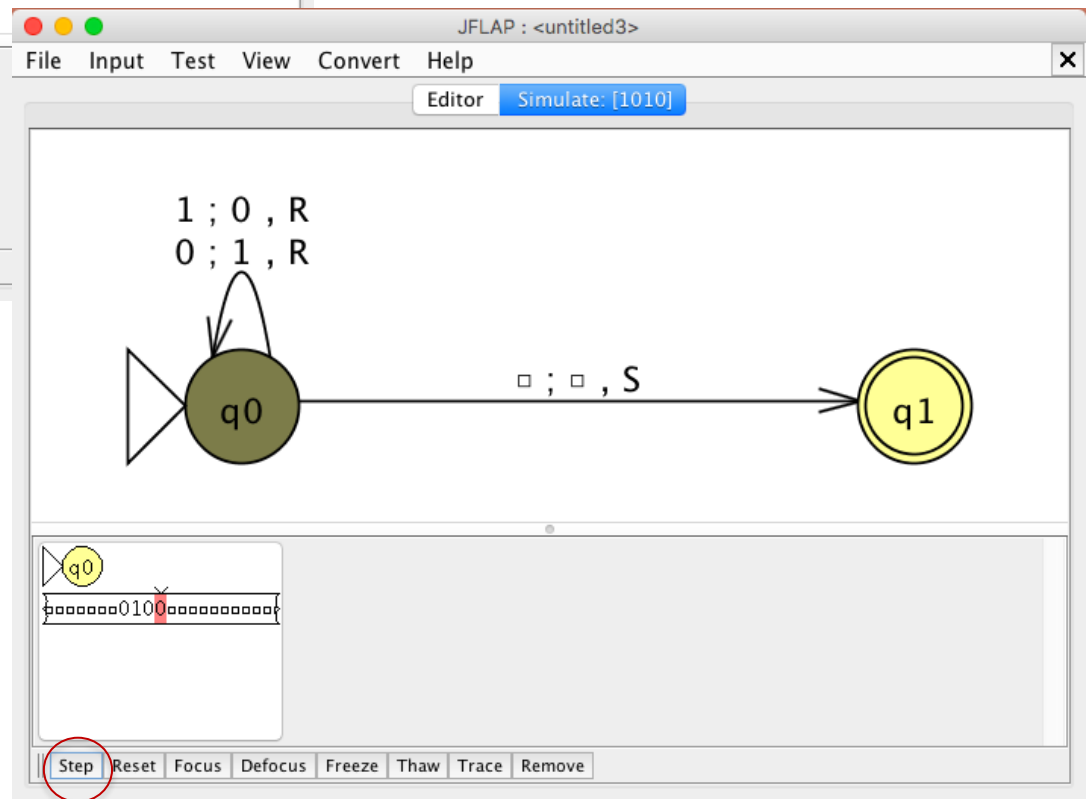
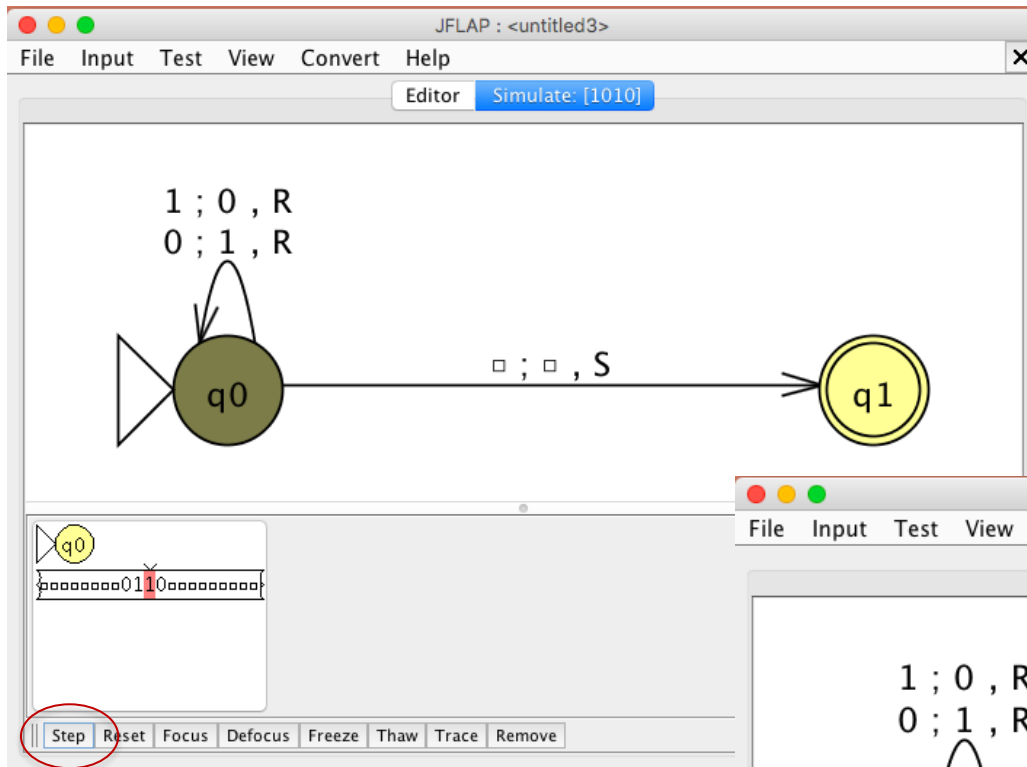
```

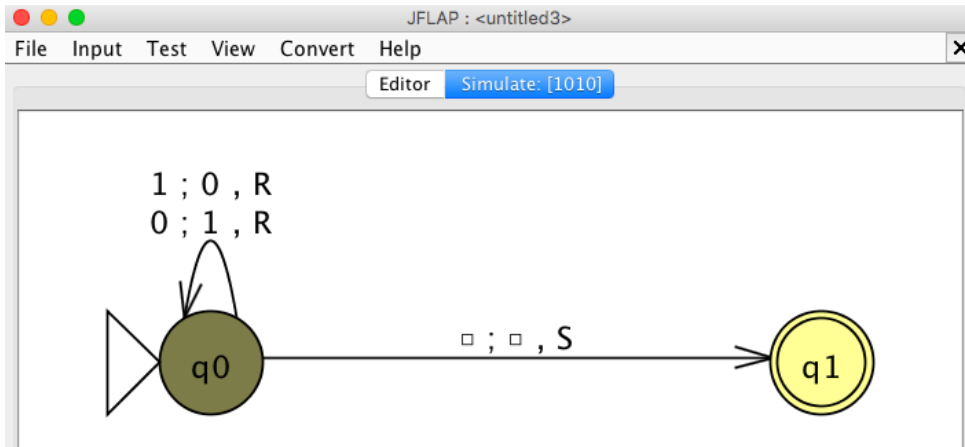
graph LR
    start(( )) --> q0((q0))
    q0 -- "1 : 0, R" --> q0
    q0 -- "0 : 1, R" --> q0
    q0 -- "a : a, S" --> q1(((q1)))
  
```

Load Inputs Run Inputs Clear Enter Lambda View Trace









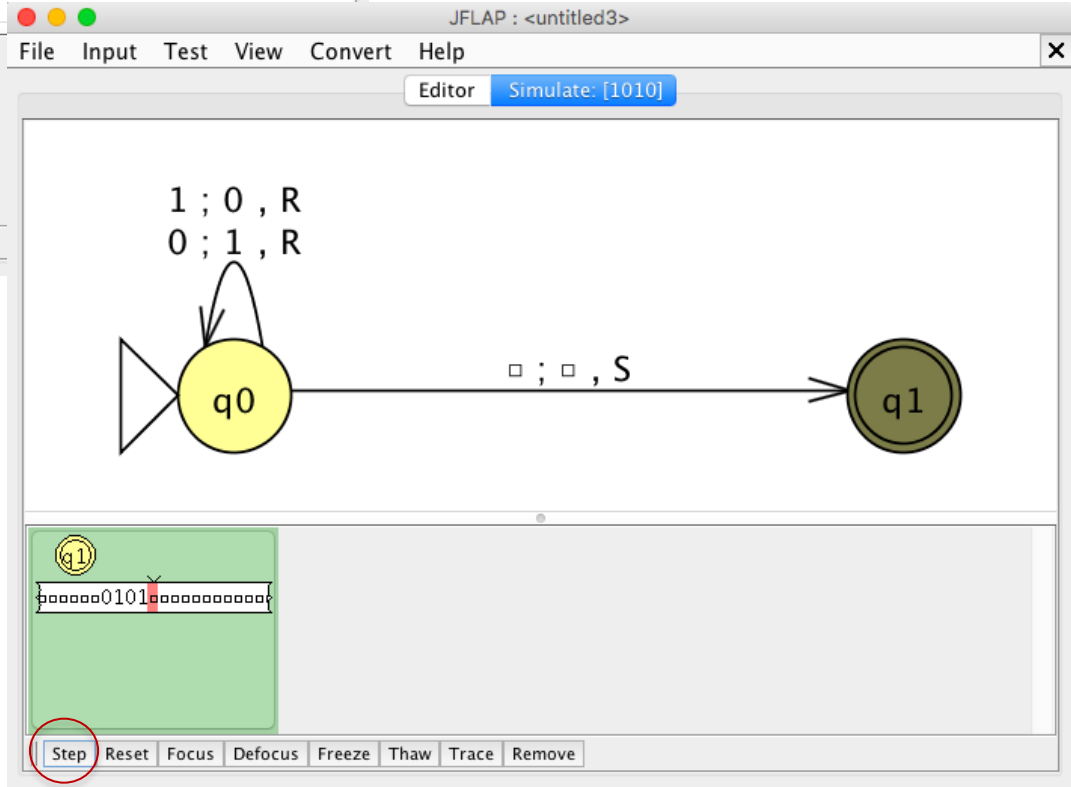
JFLAP : <untitled3>  
 File Input Test View Convert Help

Editor Simulate: [1010]

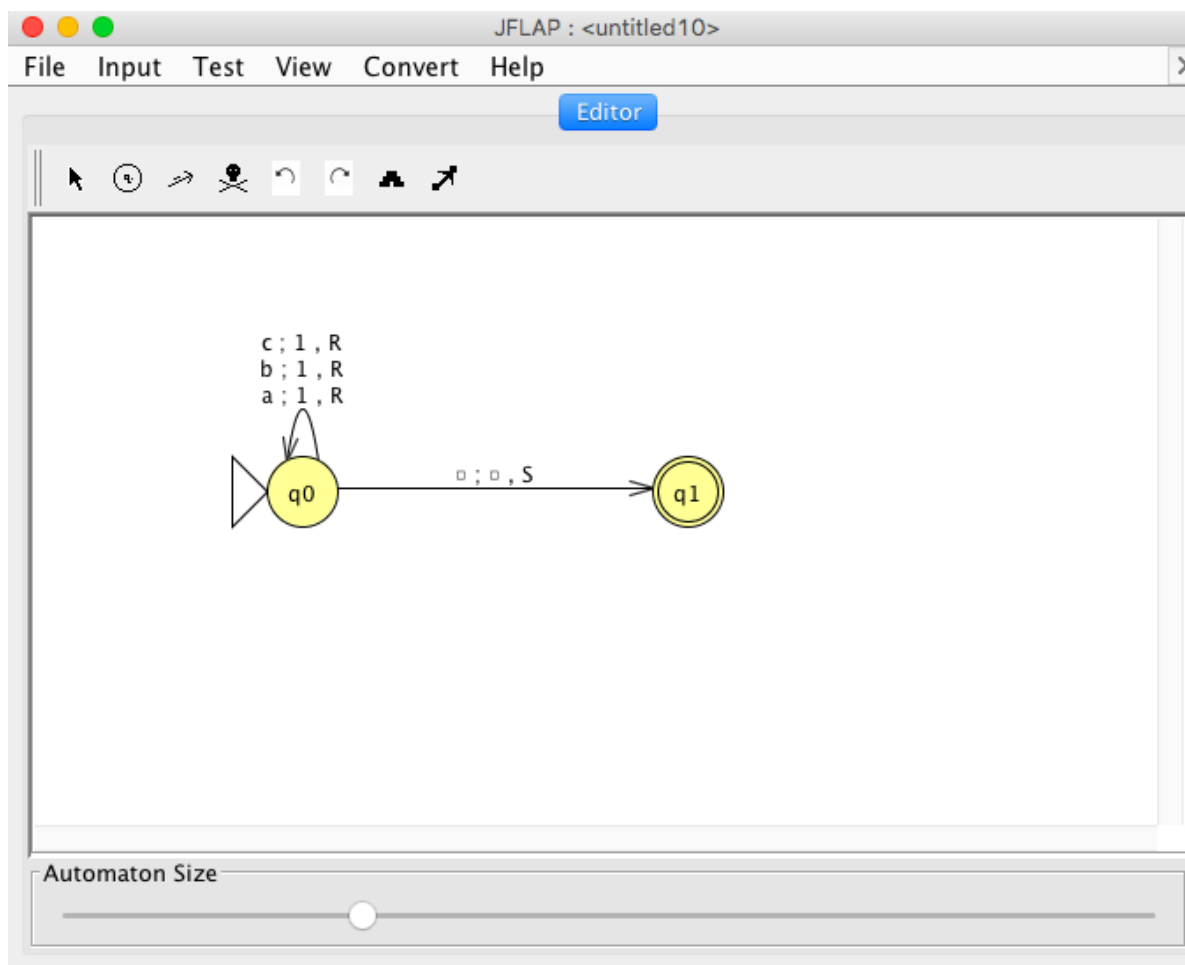
q0

0101

Step Reset Focus Defocus Freeze Thaw Trace Remove



Esercizio 2: costruire una mdT che conta il numero di caratteri presenti in una parola nell'alfabeto  $\{a,b,c\}$  (la macchina genera tanti 1 output; il numero di 1 deve essere uguale al numero di caratteri della parola in input).





JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
abc	Accept
aabbcc	Accept
abcd	Reject

$c : 1, R$   
 $b : 1, R$   
 $a : 1, R$

```

graph LR
    start(( )) --> q0((q0))
    q0 -- "a, b, c" --> q0
    q0 -- "ε" --> q1((q1))
  
```

Load Inputs Run Inputs Clear Enter Lambda View Trace

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
abc	Accept
aabbcc	Accept
abcd	Reject

$c : 1, R$   
 $b : 1, R$   
 $a : 1, R$

```

graph LR
    start(( )) --> q0((q0))
    q0 -- "a, b, c" --> q0
    q0 -- "ε" --> q1((q1))
  
```

Load Inputs Run Inputs Clear Enter Lambda View Trace

Traceback

$q1$   
 111

Load Inputs Run Inputs Clear Enter Lambda View Trace

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
abc	Accept
aabbcc	Accept
abcd	Reject

Traceback

q1

.....11111.....

c : 1, R  
b : 1, R  
a : 1, R

q0 → q1 (labeled with □ : □, S)

Load Inputs Run Inputs Clear

JFLAP : <untitled10>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
abc	Accept
aabbcc	Accept
abcd	Reject

Traceback

q0

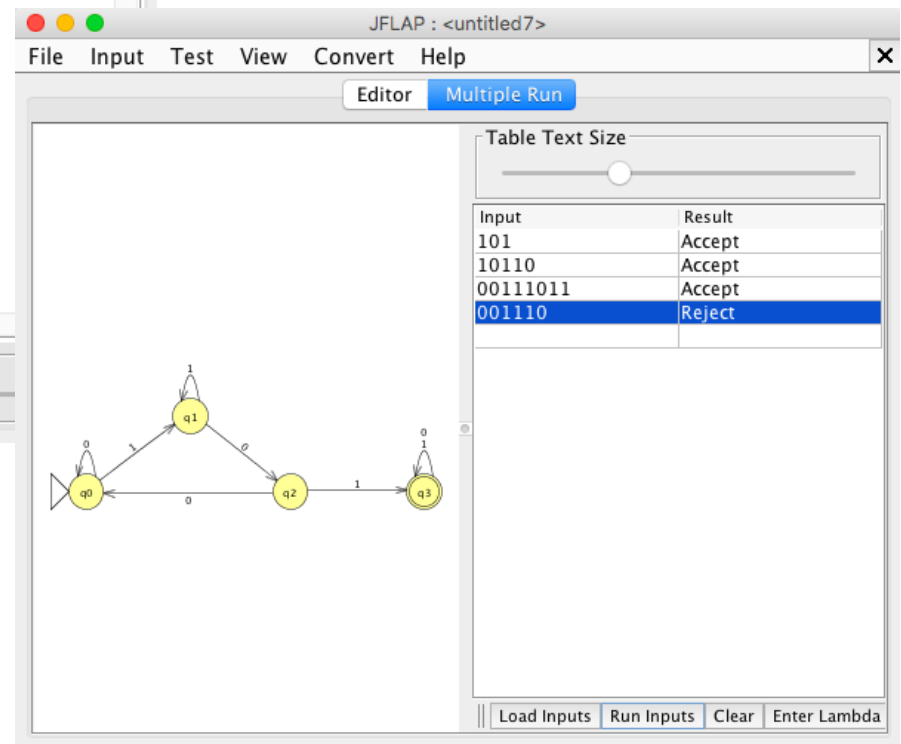
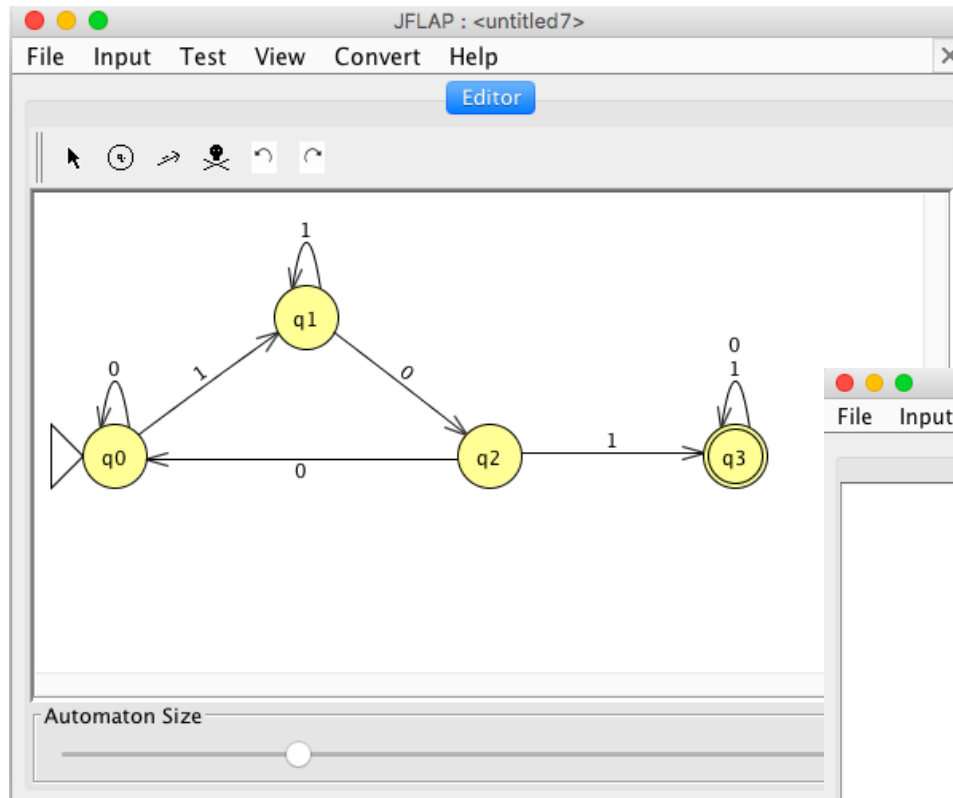
.....111.....

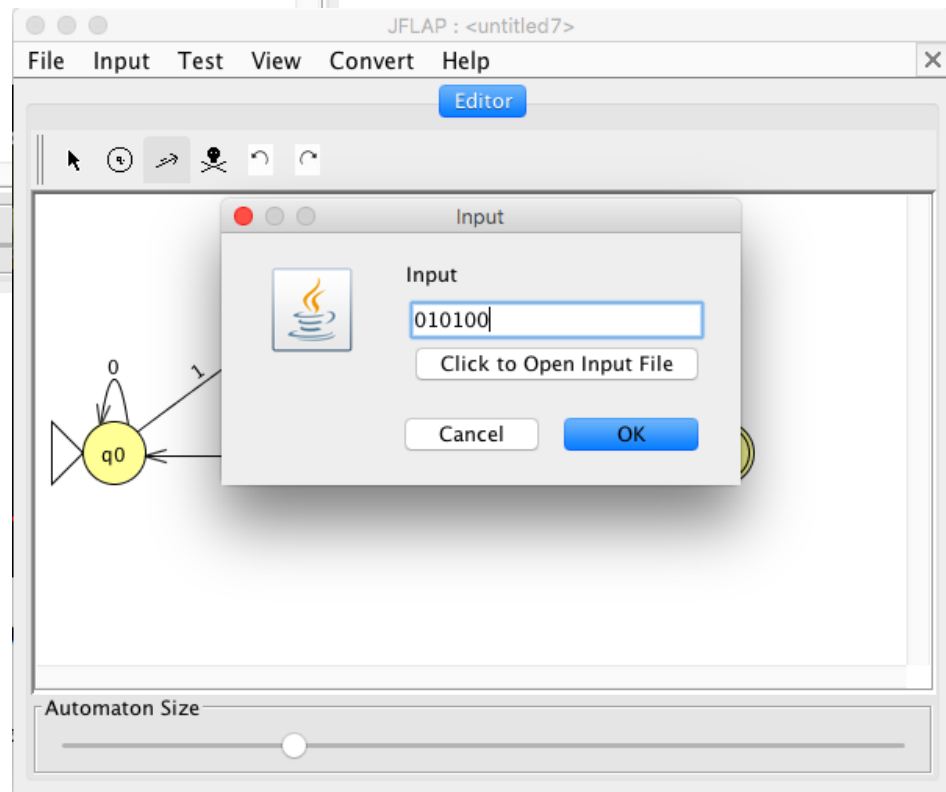
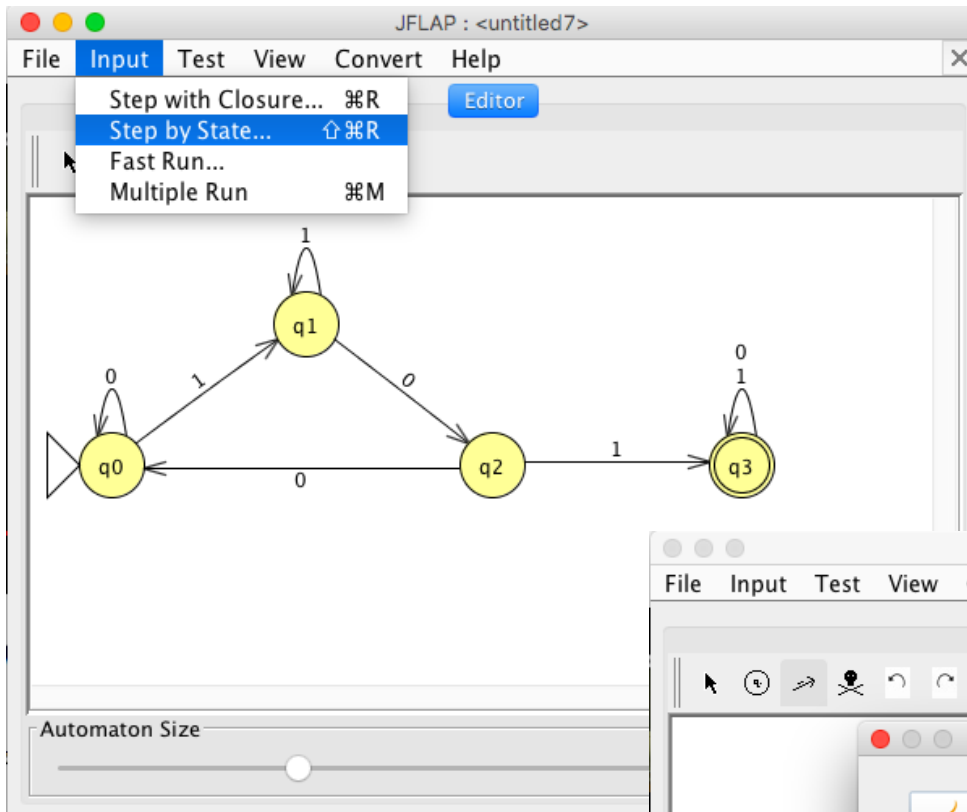
c : 1, R  
b : 1, R  
a : 1, R

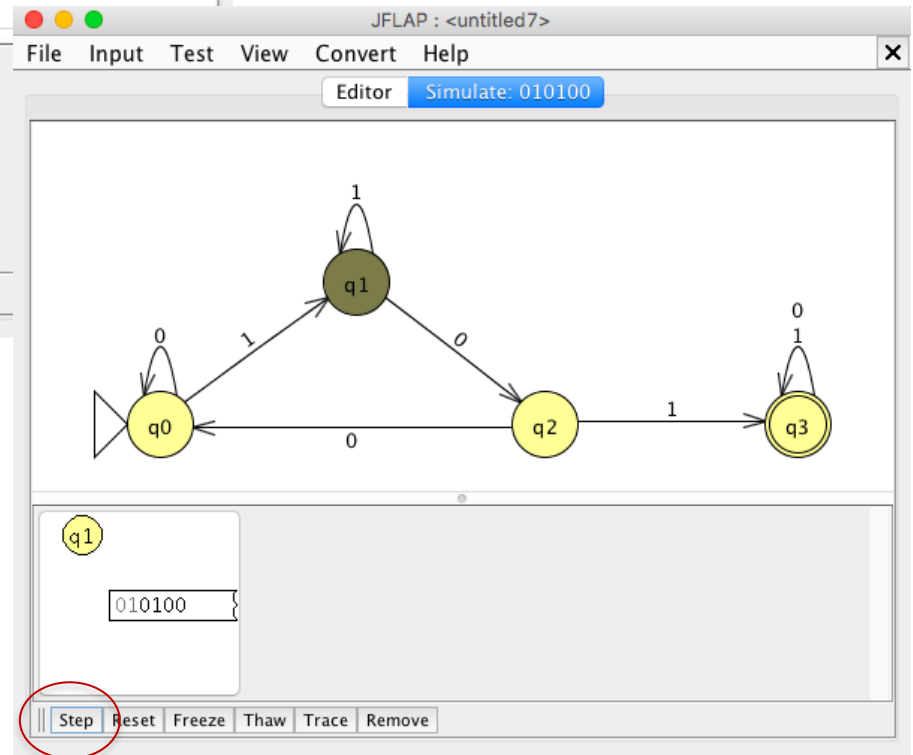
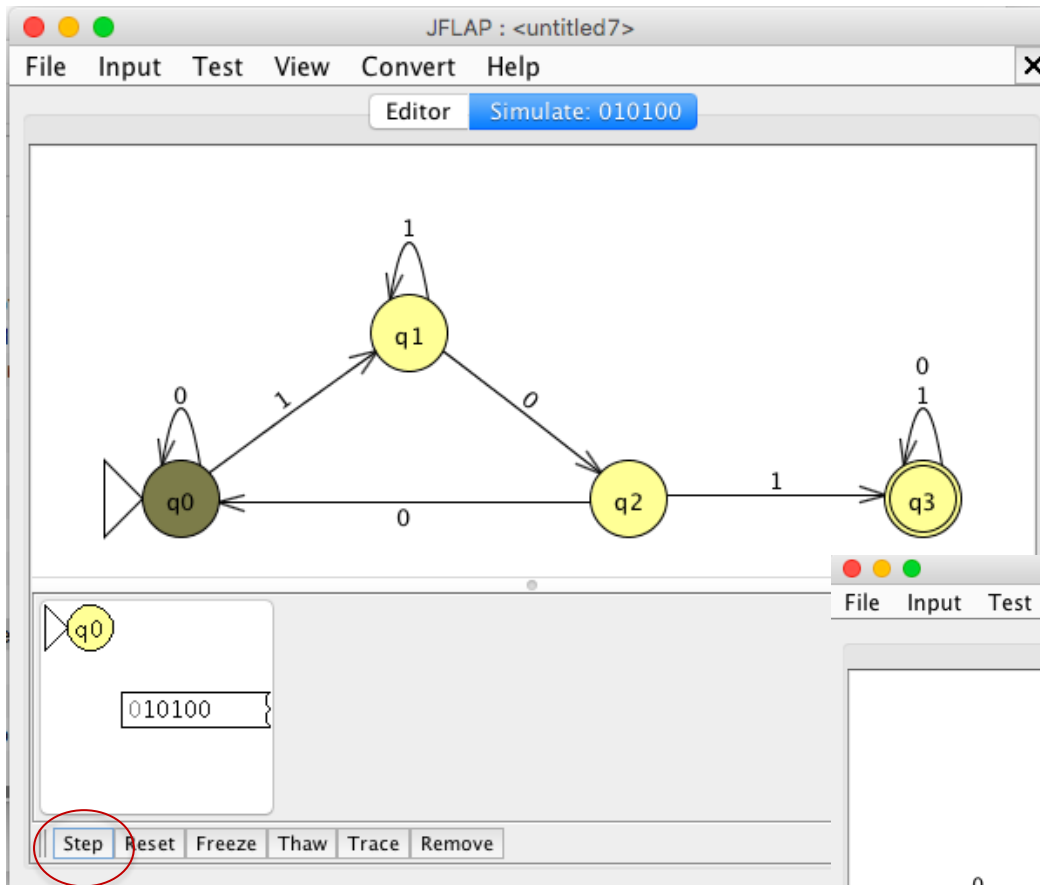
q0 → q1 (labeled with □ : □, S)

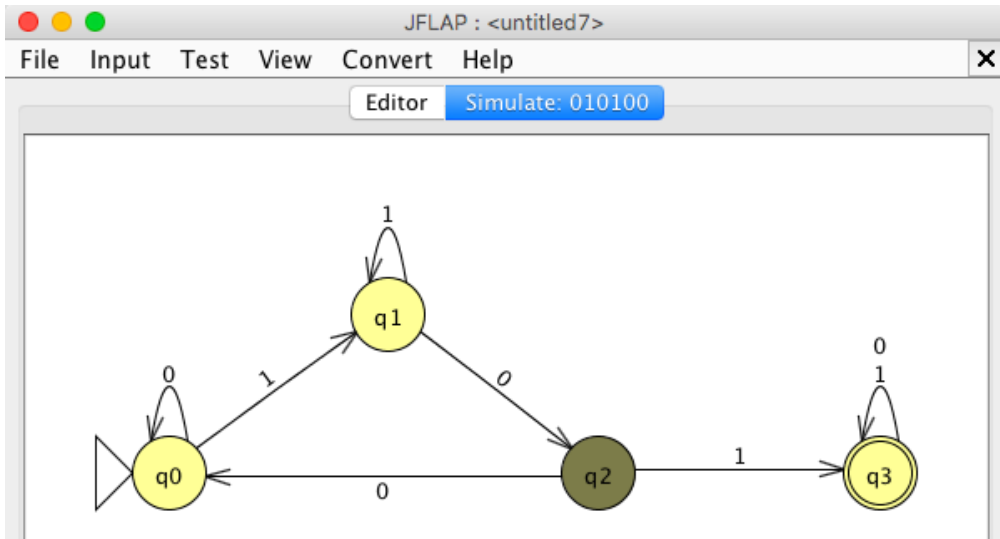
Load Inputs Run Inputs Clear Enter Lambda View Trace

Esercizio 3: costruire un automa a stati finiti che accetta stringhe che contengono '101'.





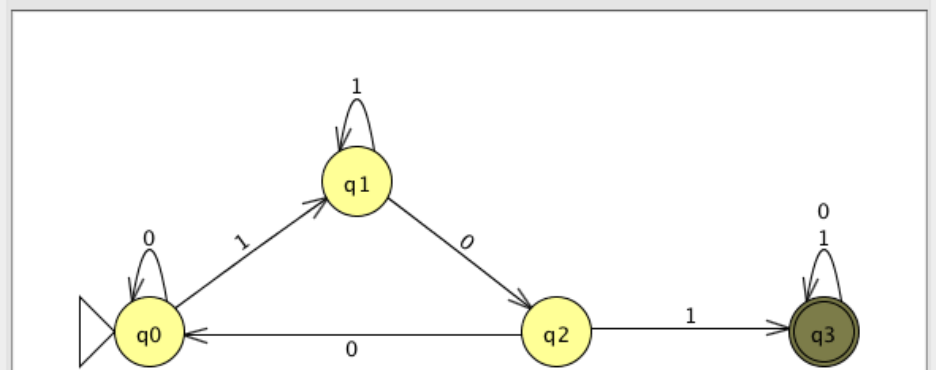
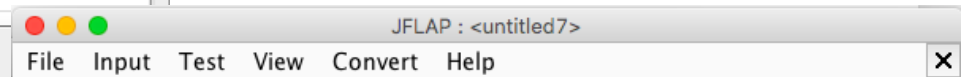




q2

010100

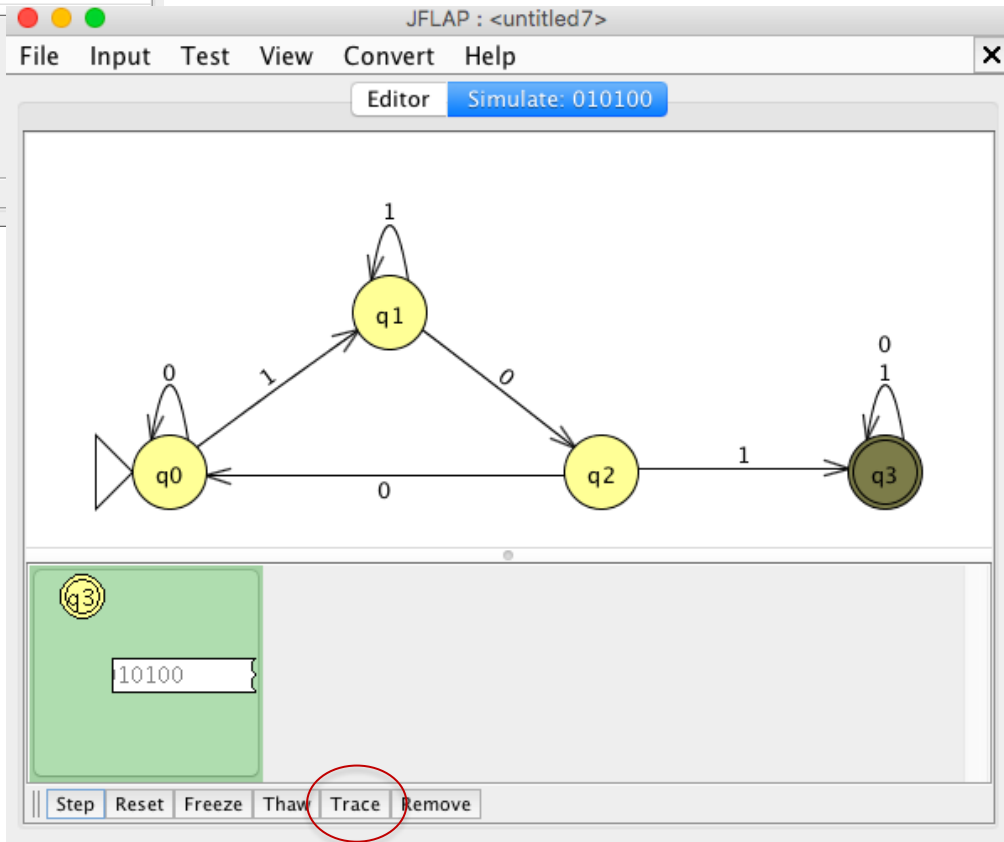
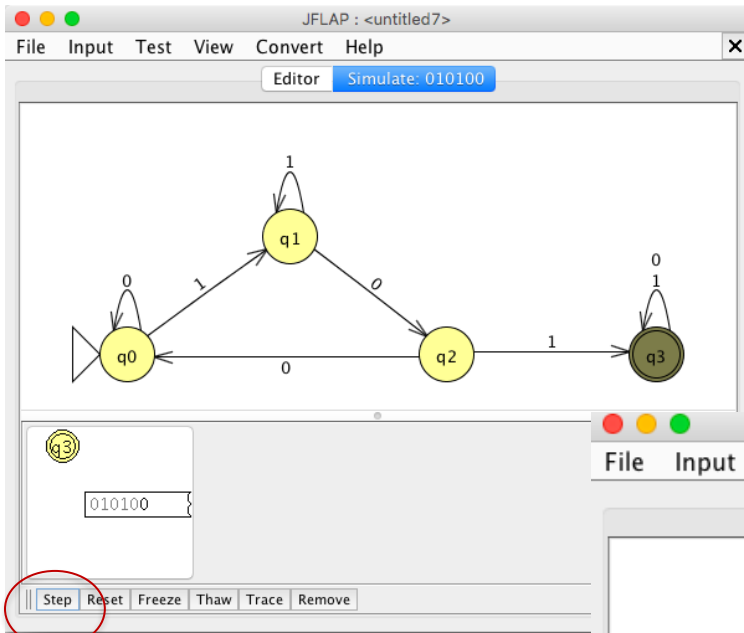
Step Reset Freeze Thaw Trace Remove



q3

010100

Step Reset Freeze Thaw Trace Remove



JFLAP : <untitled7>

File Input Test View Convert Help

The image shows the JFLAP software interface. On the left, a state transition diagram features a single state  $q_0$  with a self-loop labeled '0' and a transition labeled '1' that leads to a state that is not fully visible. Below the diagram is a stack window with a green background, containing the state  $q_3$  and the string '10100'. On the right, a 'Traceback' window displays a sequence of seven steps, each with a state label and the string '010100'. The state labels are  $q_0$ ,  $q_0$ ,  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_3$ , and  $q_3$ . The  $q_3$  labels are circled in yellow. Downward arrows connect the steps. At the bottom, a control bar includes buttons for Step, Reset, Freeze, Thaw, Trace, and Remove.

Traceback

$q_0$  010100

$q_0$  010100

$q_1$  010100

$q_2$  010100

$q_3$  010100

$q_3$  010100

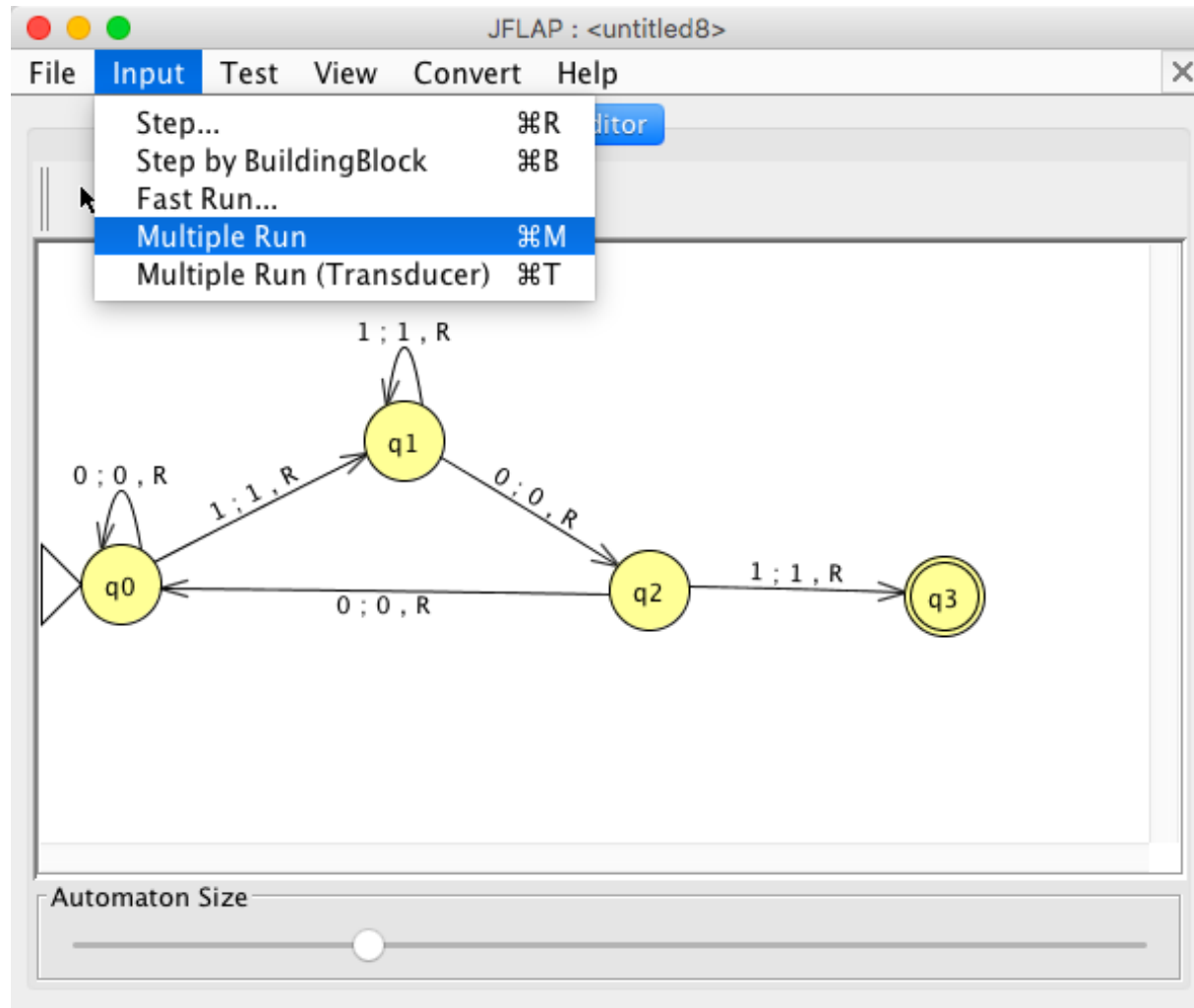
$q_3$  010100

$q_3$  010100

Step Reset Freeze Thaw Trace Remove



Esercizio 3 bis: costruire una mdT per lo stesso linguaggio dell'esercizio 3.



JFLAP : <untitled8>

File Input Test View Convert Help

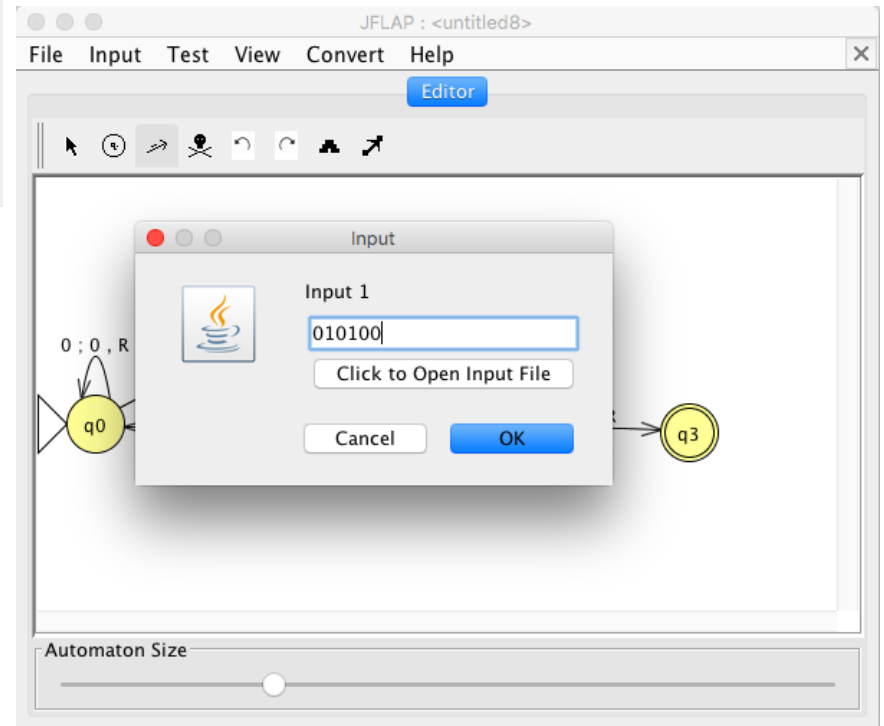
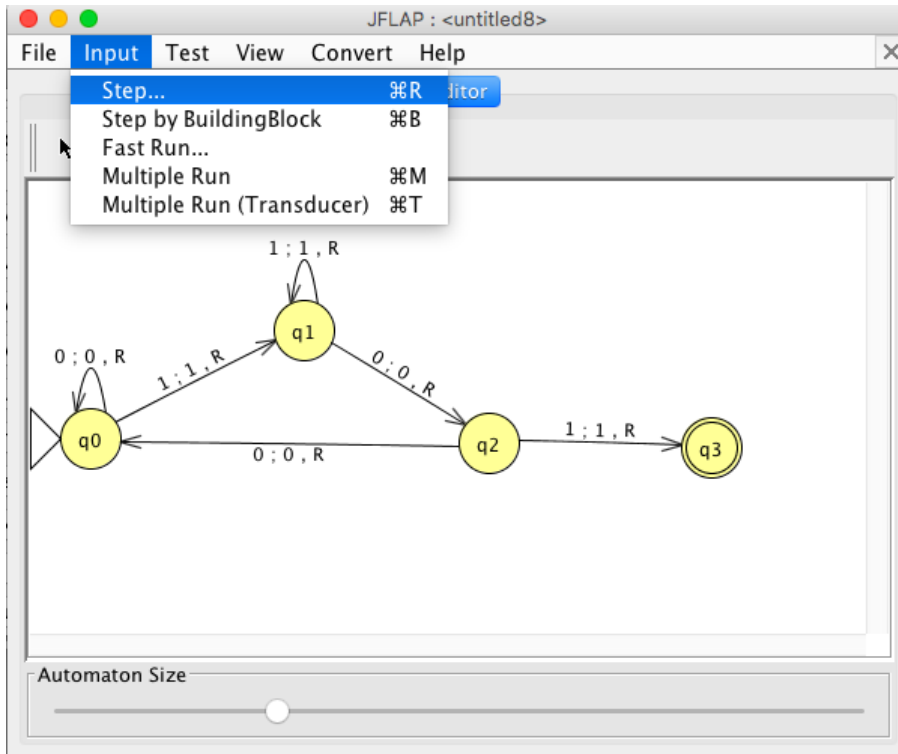
Editor Multiple Run

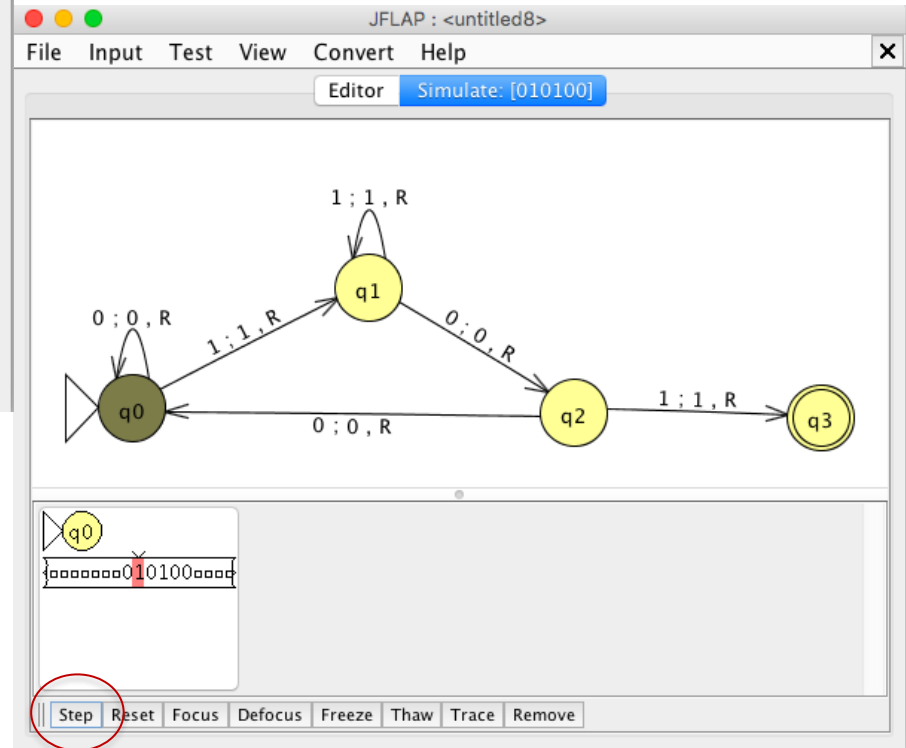
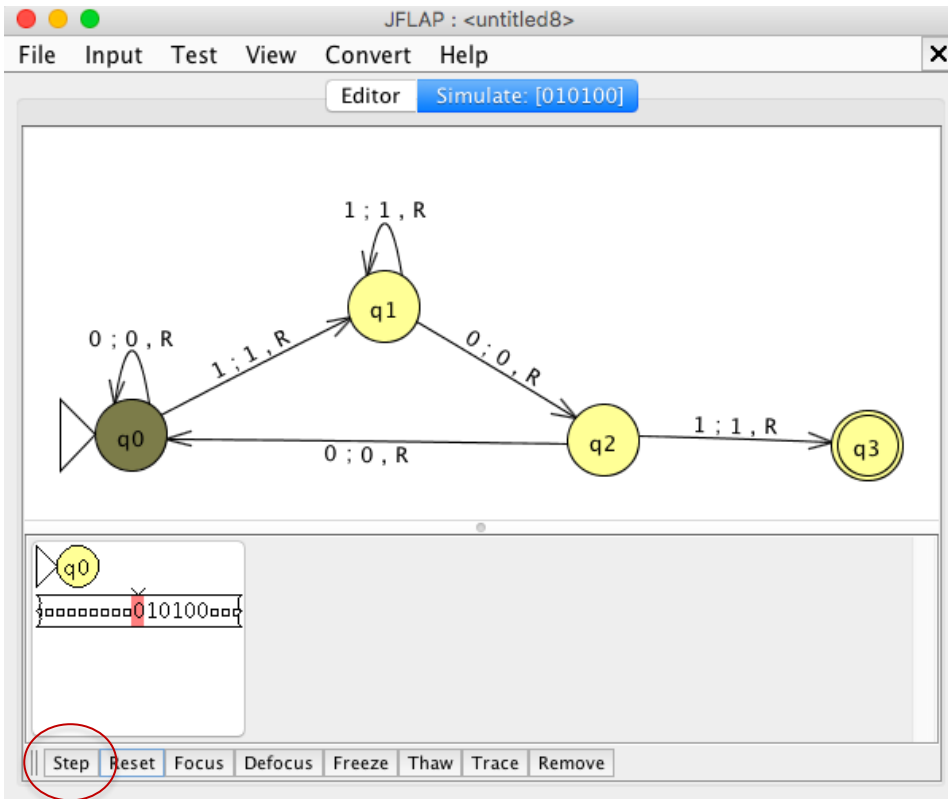
Table Text Size

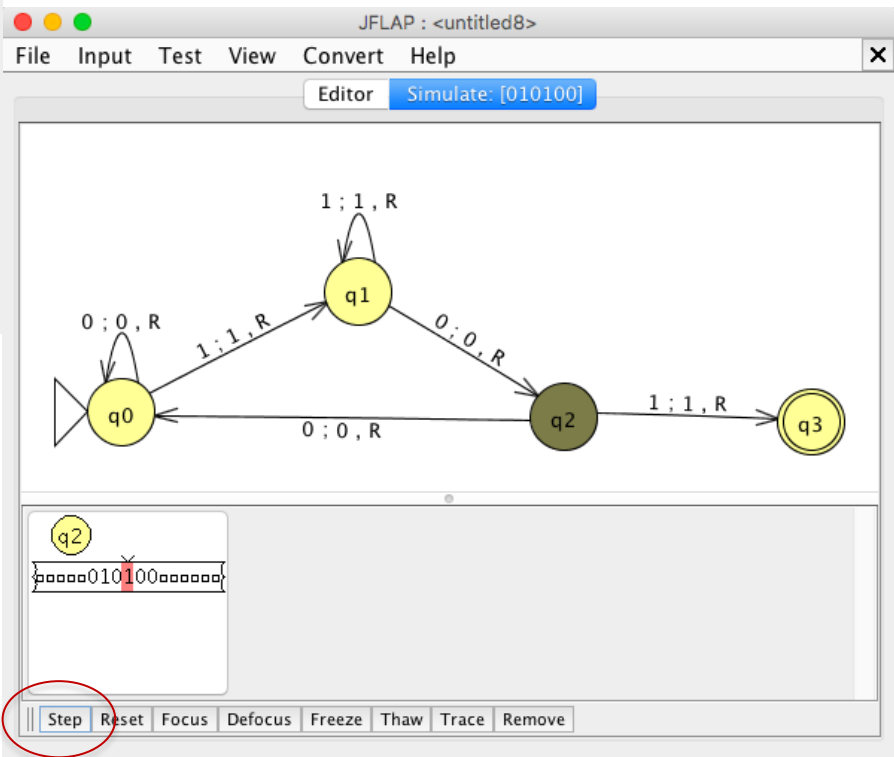
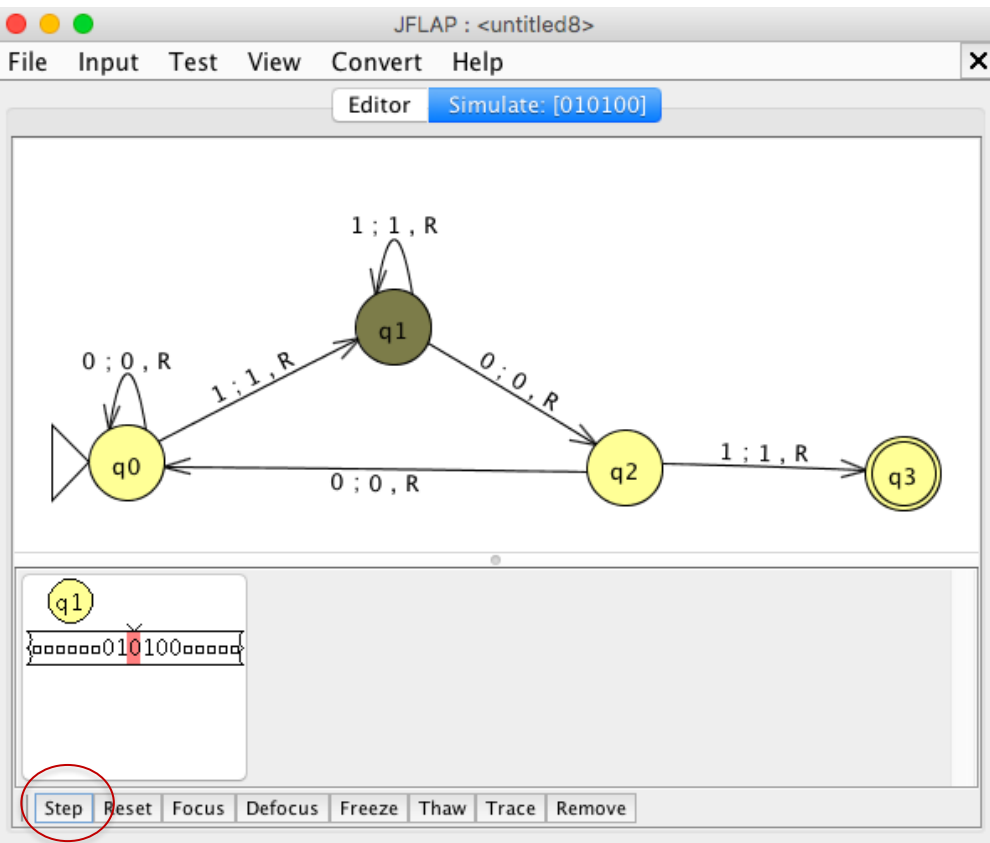
```
graph LR; q0((q0)) -- "0:0,R" --> q0; q0 -- "1:1,R" --> q1((q1)); q1 -- "1:1,R" --> q1; q1 -- "0:0,R" --> q2((q2)); q2 -- "0:0,R" --> q0; q2 -- "1:1,R" --> q3(((q3)))
```

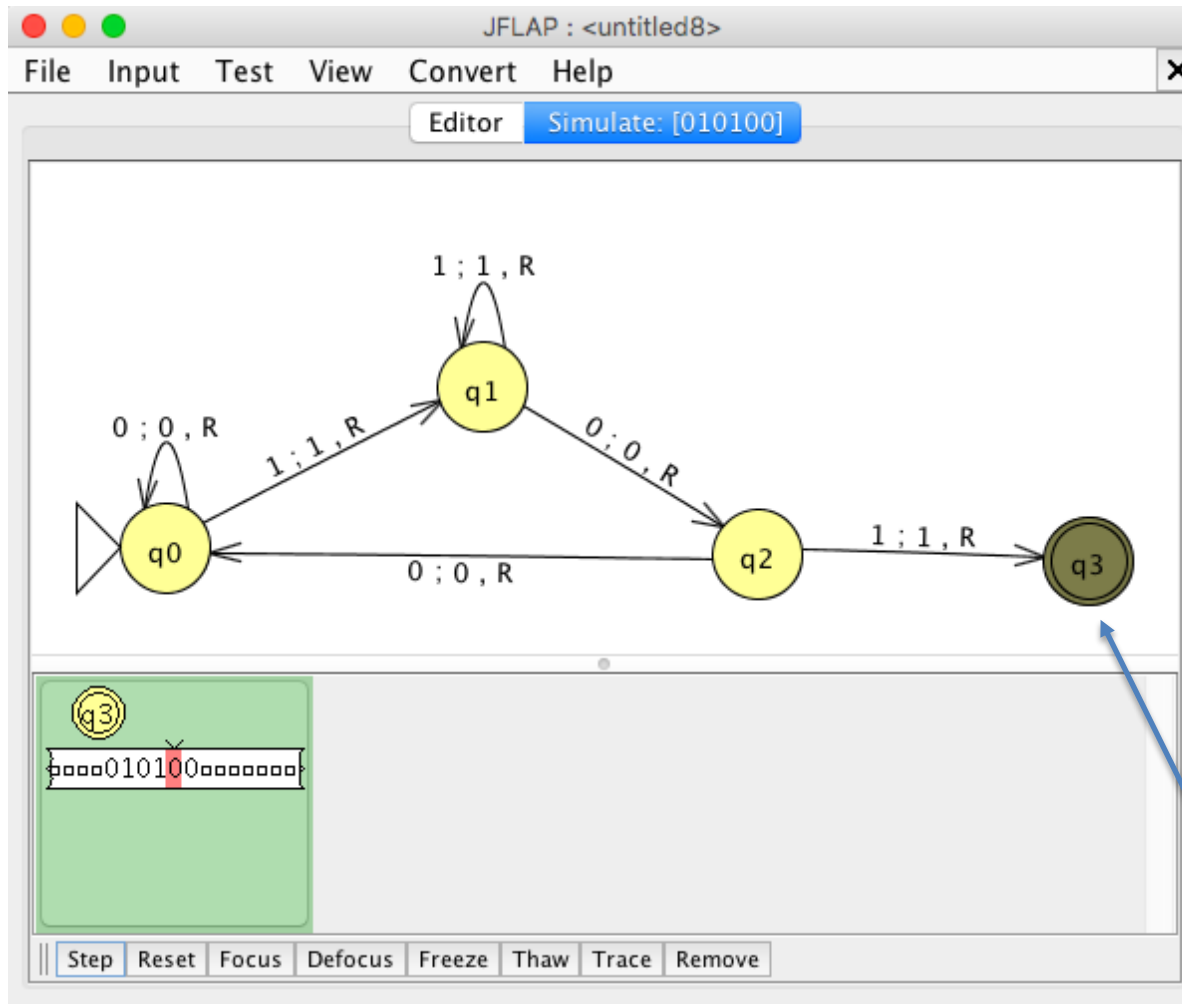
Input	Result
101	Accept
10110	Accept
00111011	Accept
001110	Reject

Load Inputs Run Inputs Clear Enter Lambda

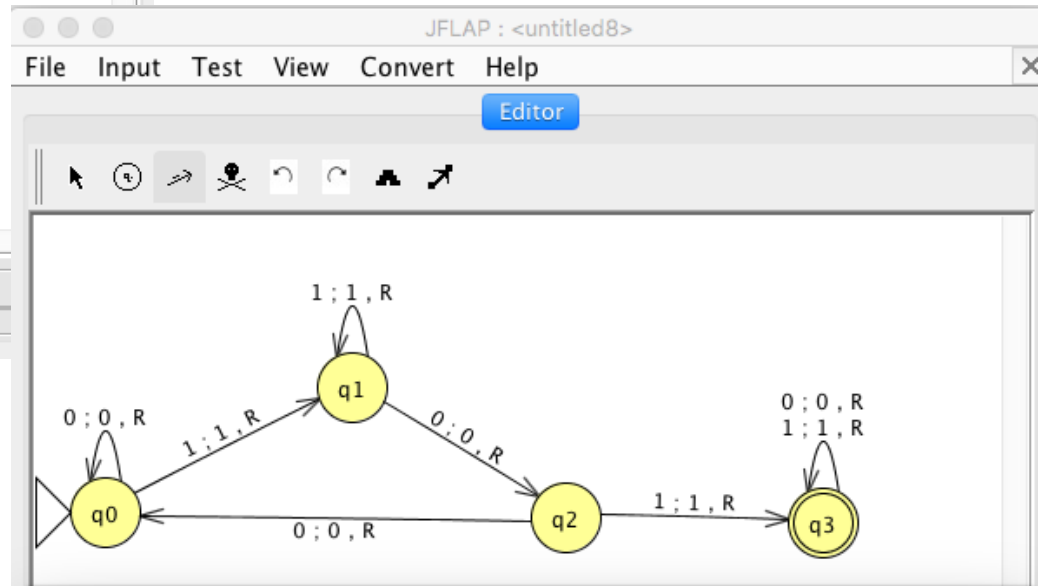
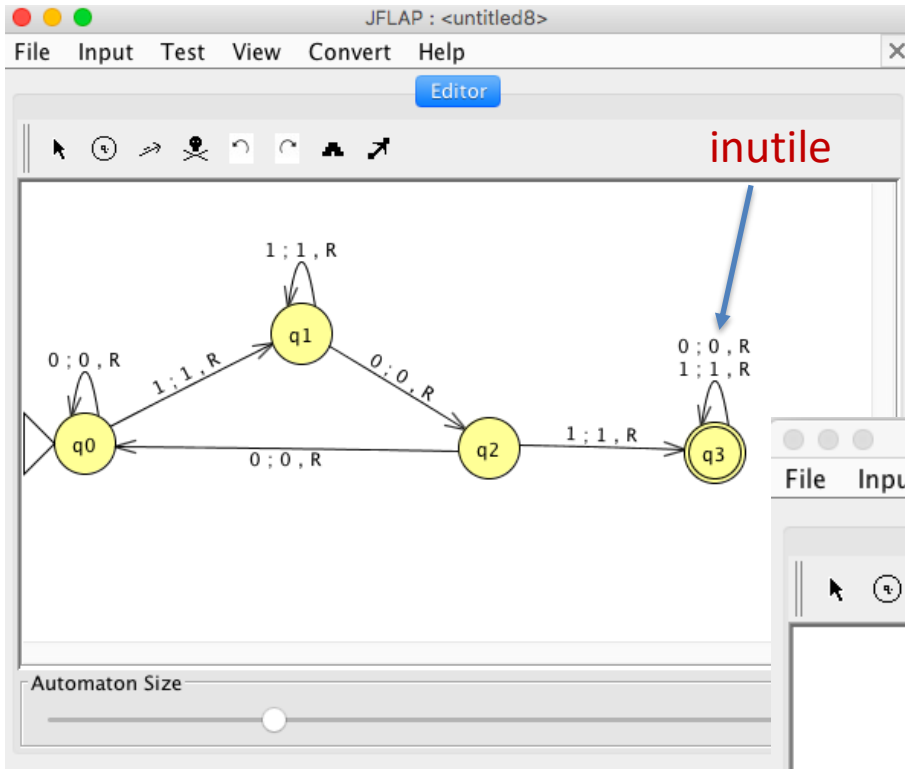








Accetta non appena la stringa contiene '101'.



Transitions From Final States

There are transitions from final states. Please remove them or change the preference in the "Preferences" menu in the JFLAP main menu.

OK

Esercizio 4: costruire una mdT che accetta il linguaggio

$$L = \{a^n b^n \mid n > 0\}$$

~~a~~ a a a b b b ~~b~~

x ~~a~~ a a b b ~~b~~ y

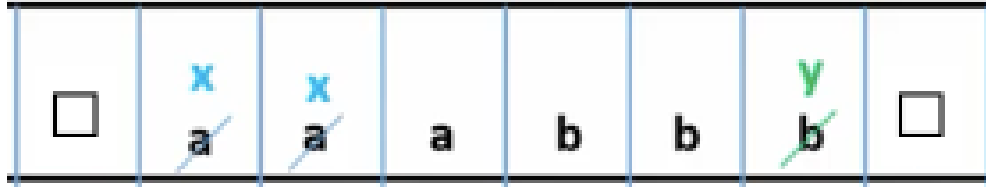
x x ~~a~~ a b ~~b~~ y y

x x x ~~a~~ ~~b~~ y y y

x x x x y y y y

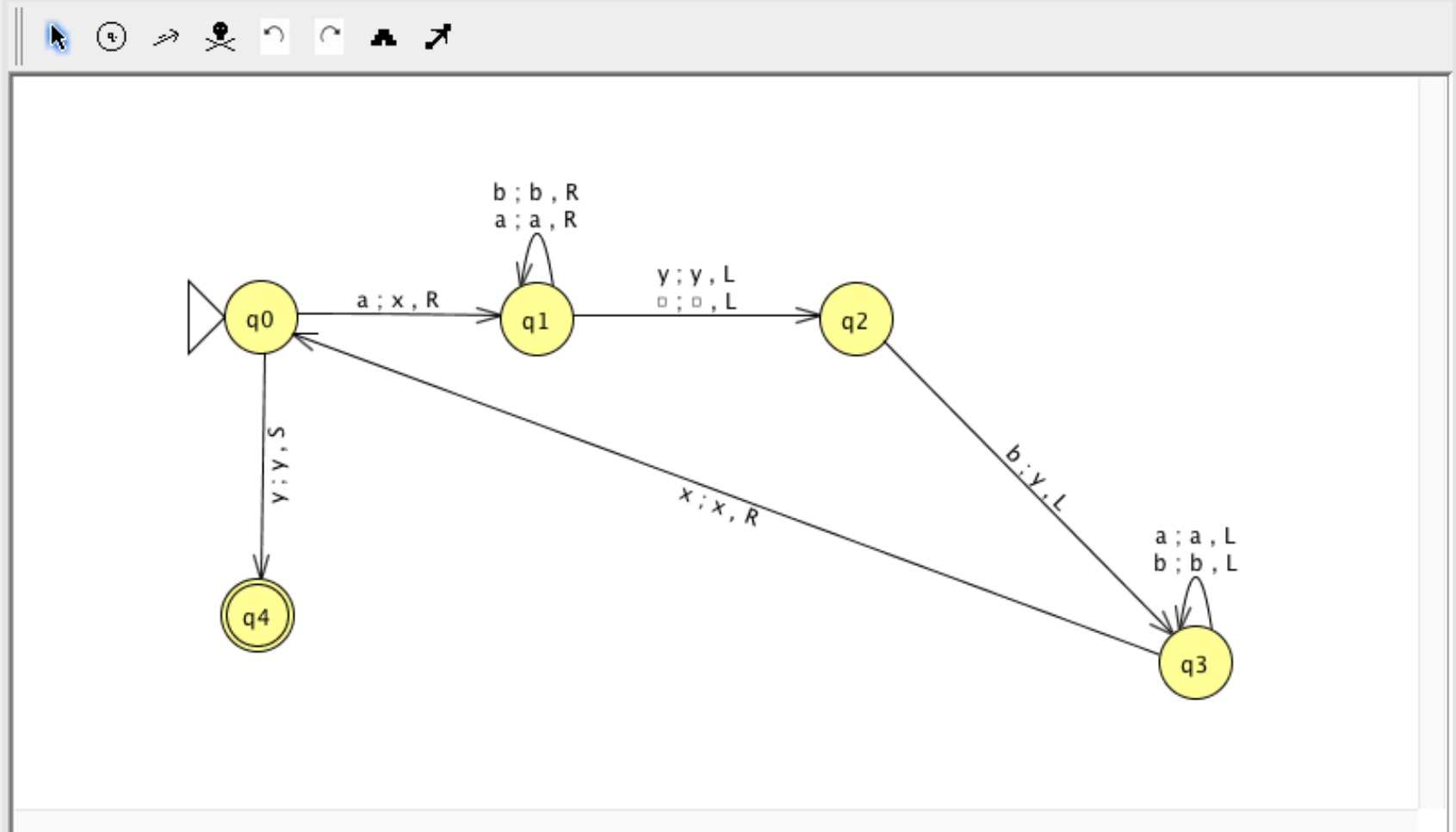


## Turing Machine for $a^n b^n$

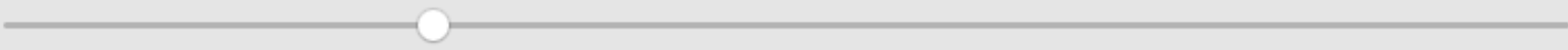


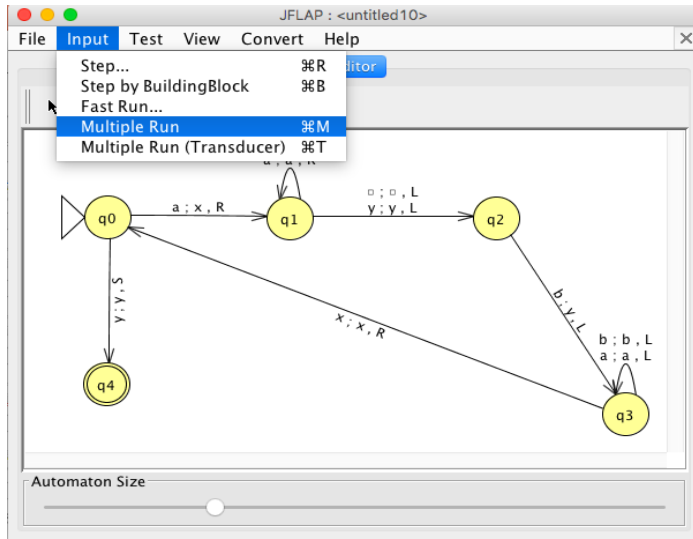
Si ferma quando cerchiamo una 'a' e troviamo una 'y'.

1. Leggere una 'a' e sostituire con una 'x'
2. Saltare a destra tutte le 'a' e le 'b' fino al blank
3. Andare a sinistra
4. Leggere una 'b' e sostituire con una 'y'
5. Saltare a sinistra tutte le 'b' e le 'a' fino a una 'x'
6. Andare a destra
7. Ripetere; (in 2. e 3. cercare una 'y' invece di un blank)



Automaton Size





JFLAP : <untitled9>

ew Convert Help

Editor Multiple Run

Table Text Size

Input	Result
ab	Accept
aabb	Accept
aabbb	Reject
aaabbb	Accept
aaaabbb	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

JFLAP : <untitled9>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
ab	Accept
aabb	Accept
aaabbb	Reject
aaaabbb	Accept
aaaaabbb	Reject

Traceback

q4

Load Inputs Run

JFLAP : <untitled9>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
ab	Accept
aabb	Accept
aaabbb	Reject
aaaabbb	Accept
aaaaabbb	Reject

Traceback

q2

Load Inputs Run Inputs Clear Enter Lambda View Trace