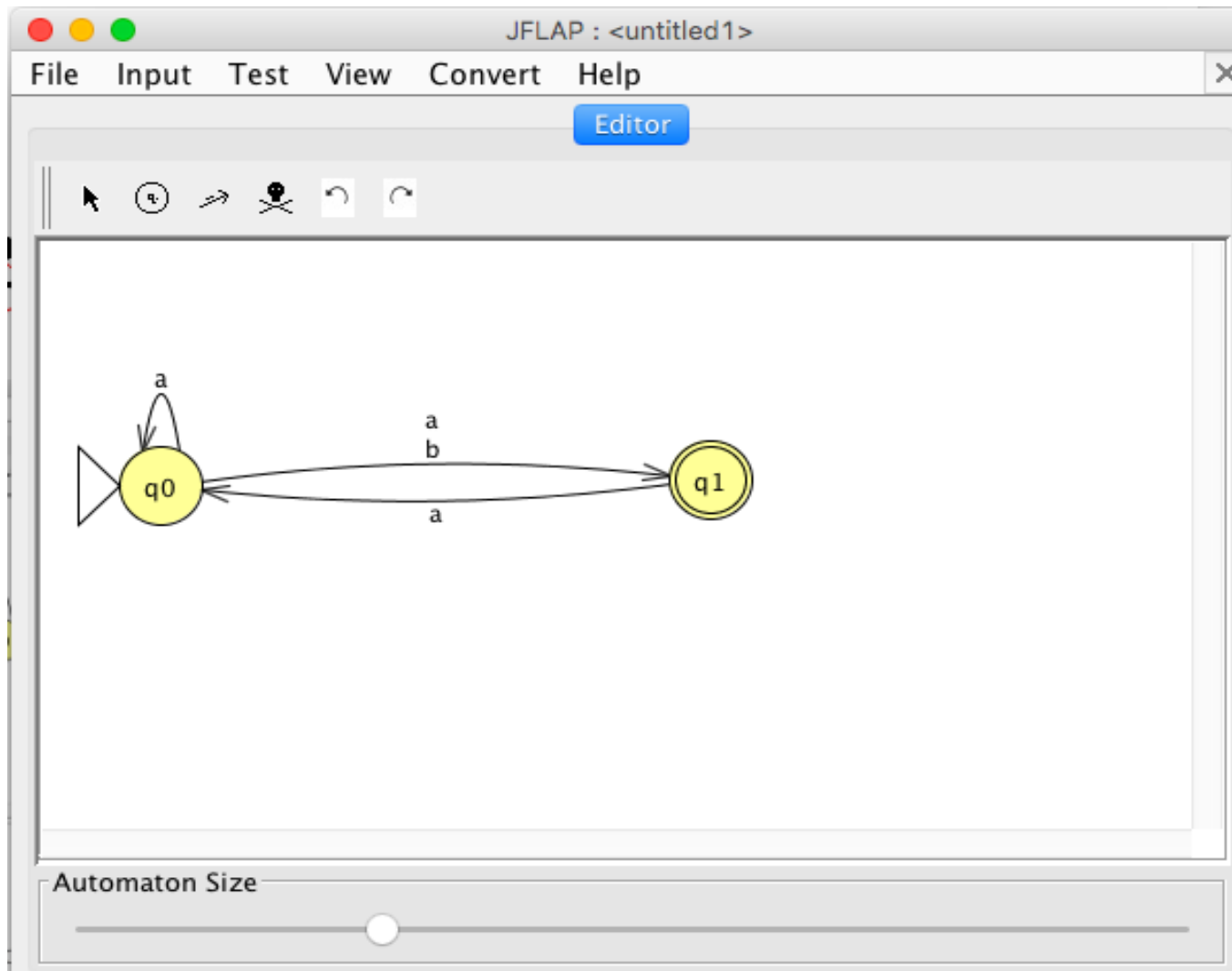
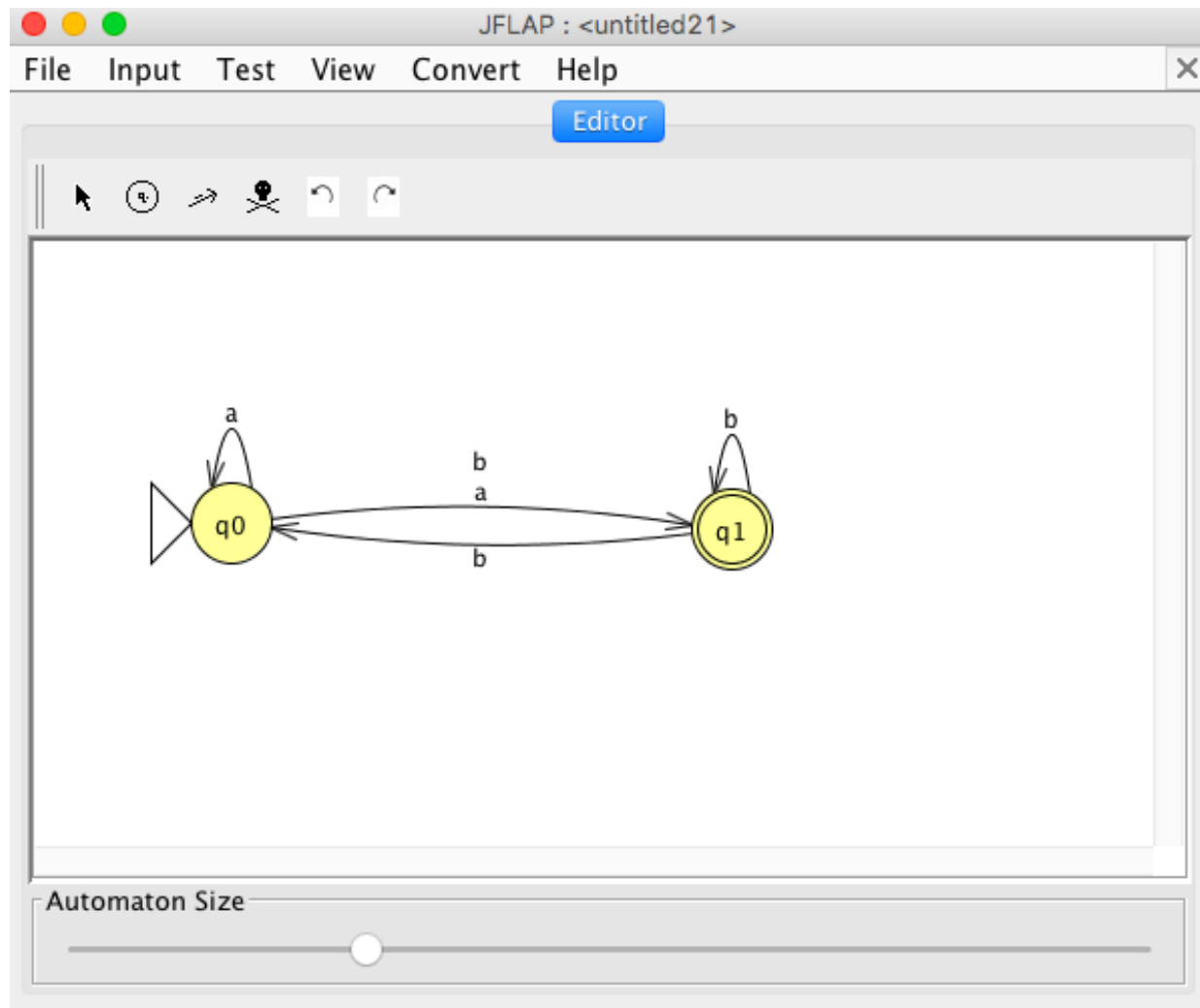


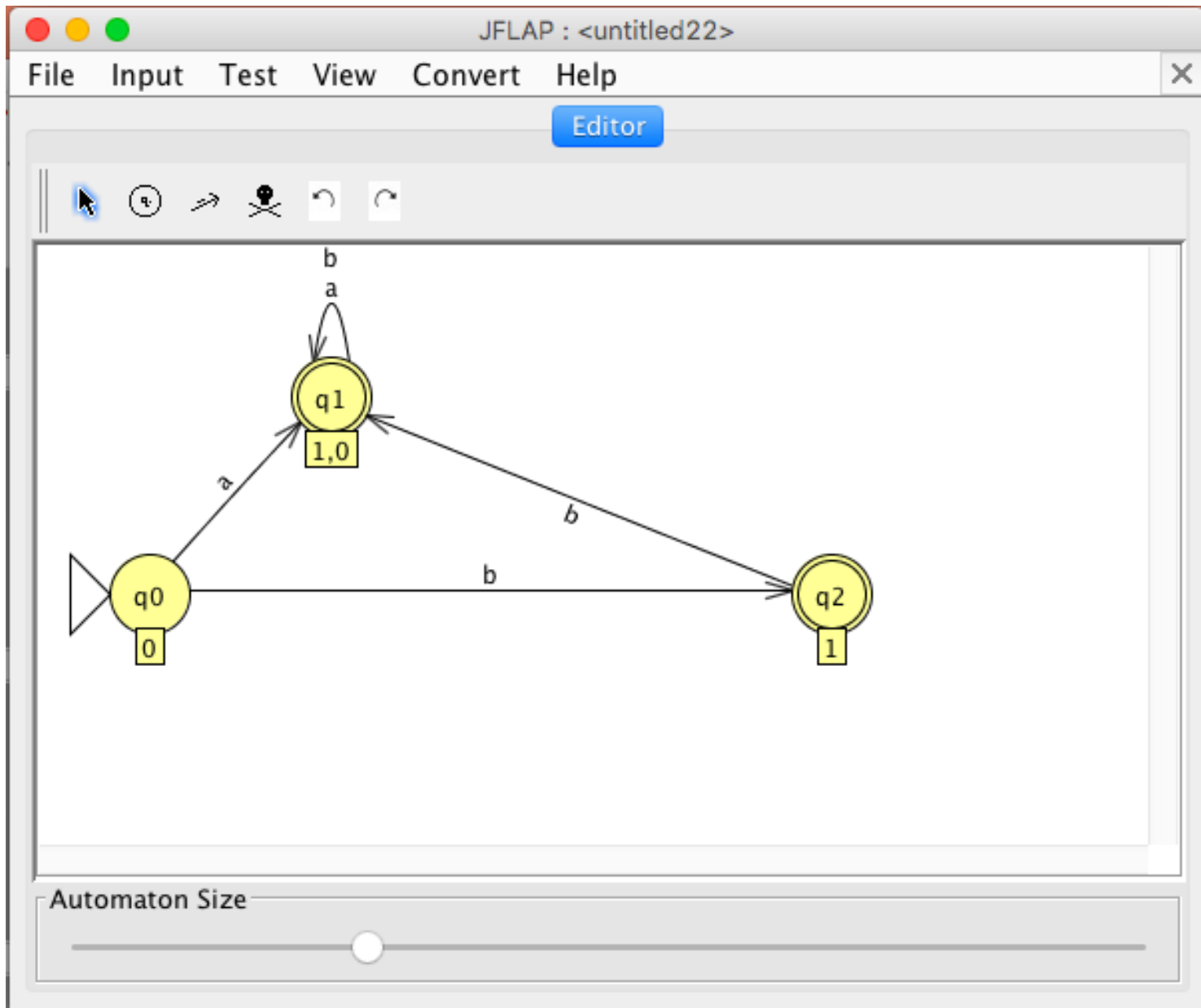
Esercizi

Esercizio 1: Costruire il DFA che accetta
 $L = \{w / w \text{ in } \{a,b\}^*, w \text{ non contiene due } b \text{ consecutive}\}$



Esercizio 2: Trasformare il seguente automa non deterministico M in automa deterministico M' equivalente





Esercizio 3: data la grammatica regolare con le seguenti produzioni:

$$S \rightarrow 0A \mid 1B \mid 0S$$
$$A \rightarrow aB \mid bA \mid a$$
$$B \rightarrow bA \mid aB \mid b$$

Generare l'NFA e il DFA che riconoscono il linguaggio generato da tale grammatica.

A partire dall'automa deterministico, generare la grammatica corrispondente.

JFLAP : <untitled1>

File Input Test Convert Help

Editor

Table Text Size

LHS		RHS
S	→	0A
S	→	1B
S	→	0S
A	→	aB
A	→	bA
A	→	a
B	→	bA
B	→	aB
B	→	b

JFLAP : <untitled1>

File Input Test Convert Help

Editor Convert to FA

Table Text Size

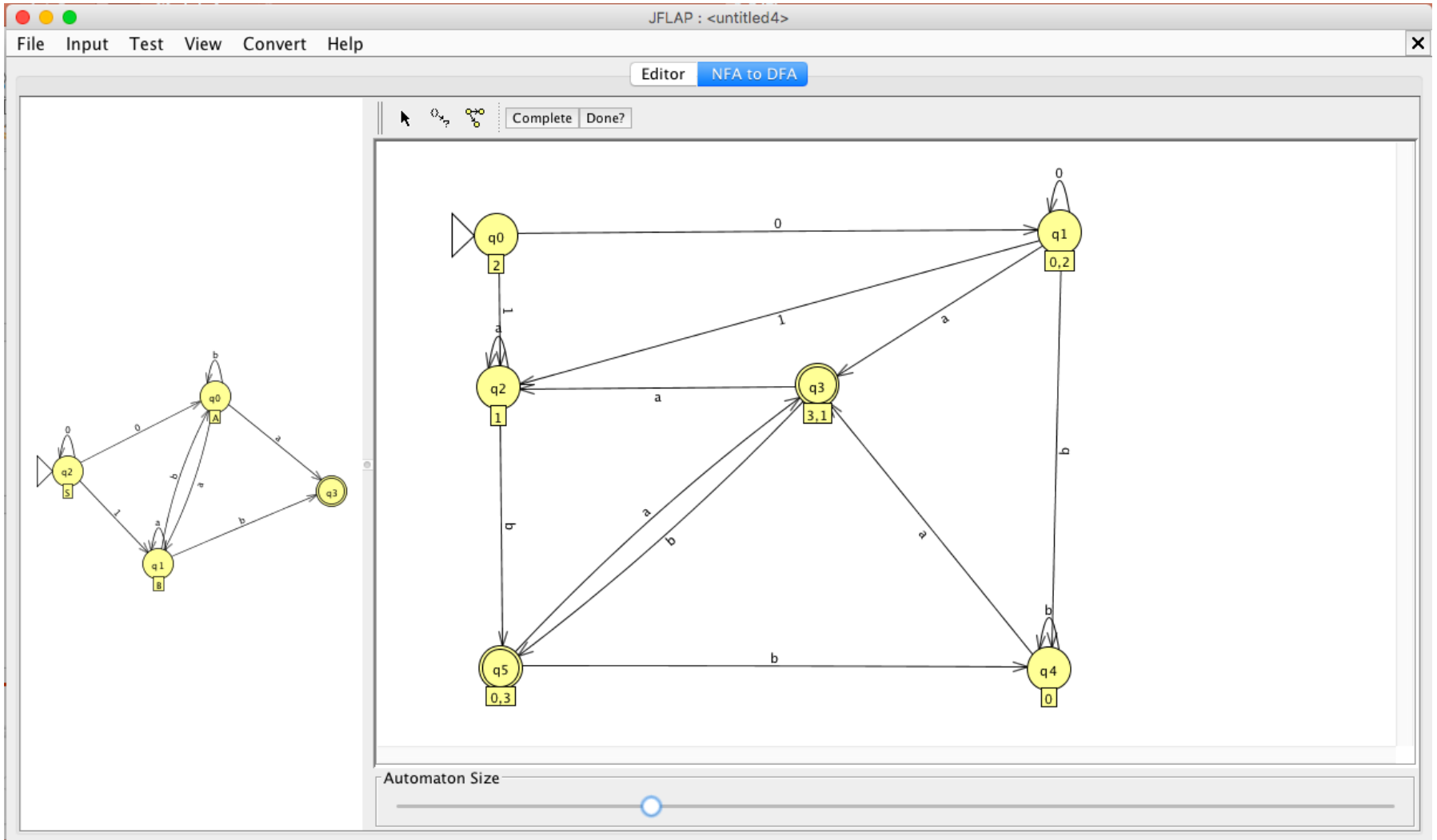
Production	Created
S→0A	✓
S→1B	✓
S→0S	✓
A→aB	✓
A→bA	✓
A→a	✓
B→bA	✓
B→aB	✓
B→b	✓

Show All Create Selected Done? Export

```

graph TD
    q2((q2)) -- 0 --> q2
    q2 -- 0 --> q0((q0))
    q2 -- 1 --> q1((q1))
    q0 -- 0 --> q0
    q0 -- a --> q1
    q0 -- a --> q3(((q3)))
    q1 -- a --> q1
    q1 -- b --> q1
    q1 -- b --> q3
    q1 -- a --> q0
    q1 -- b --> q0
  
```

Automaton Size



JFLAP : <untitled8>

File Input Test View Convert Help

Editor Convert to Grammar

Hint Show All What's Left? Export

LHS	RHS
E	→ aC
C	→ aB
B	→ aB
D	→ bD
B	→ bE
C	→ bE
S	→ 1B
A	→ bD
S	→ 0A
D	→ aC
E	→ bD
A	→ aC
A	→ 0A
E	→ λ
C	→ λ
A	→ 1B

Table Text Size

Esercizio 4: Dati due automi deterministici

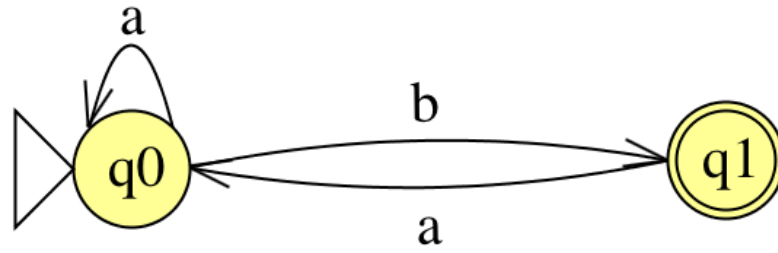
$$A_1=(Q_1,\Sigma_1,\delta_1,S_1,F_1) \text{ e } A_2=(Q_2,\Sigma_2,\delta_2,S_2,F_2)$$

che accettano i linguaggi $L_1=L(A_1)$ e $L_2=L(A_2)$, rispettivamente

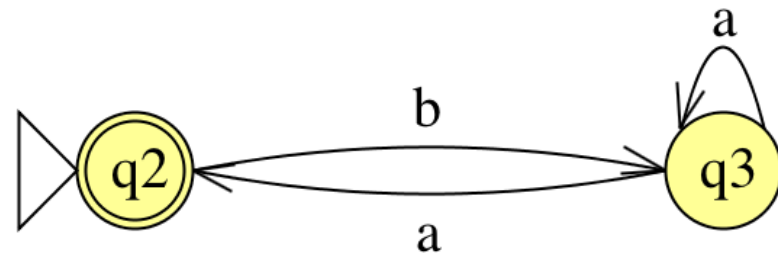
costruire un automa M che riconosce il linguaggio

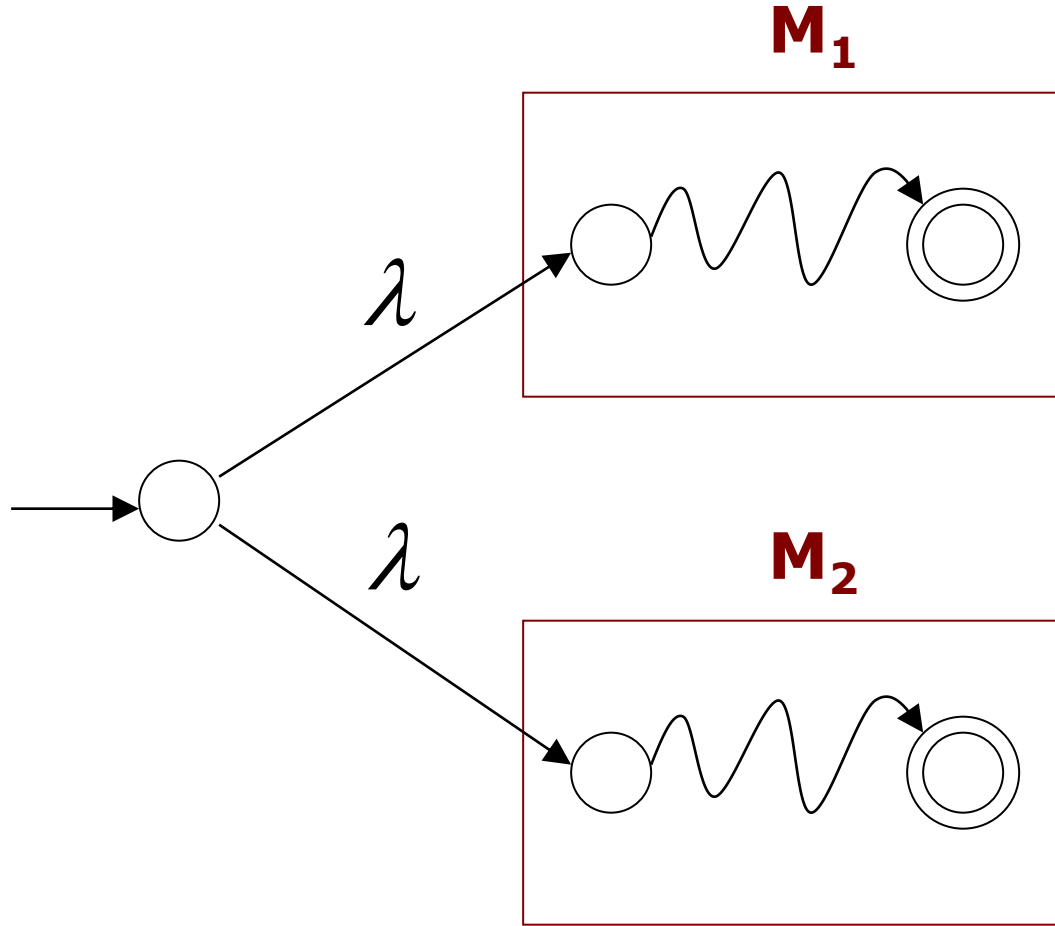
$$L=L(M)=L(M_1) \cup L(M_2)$$

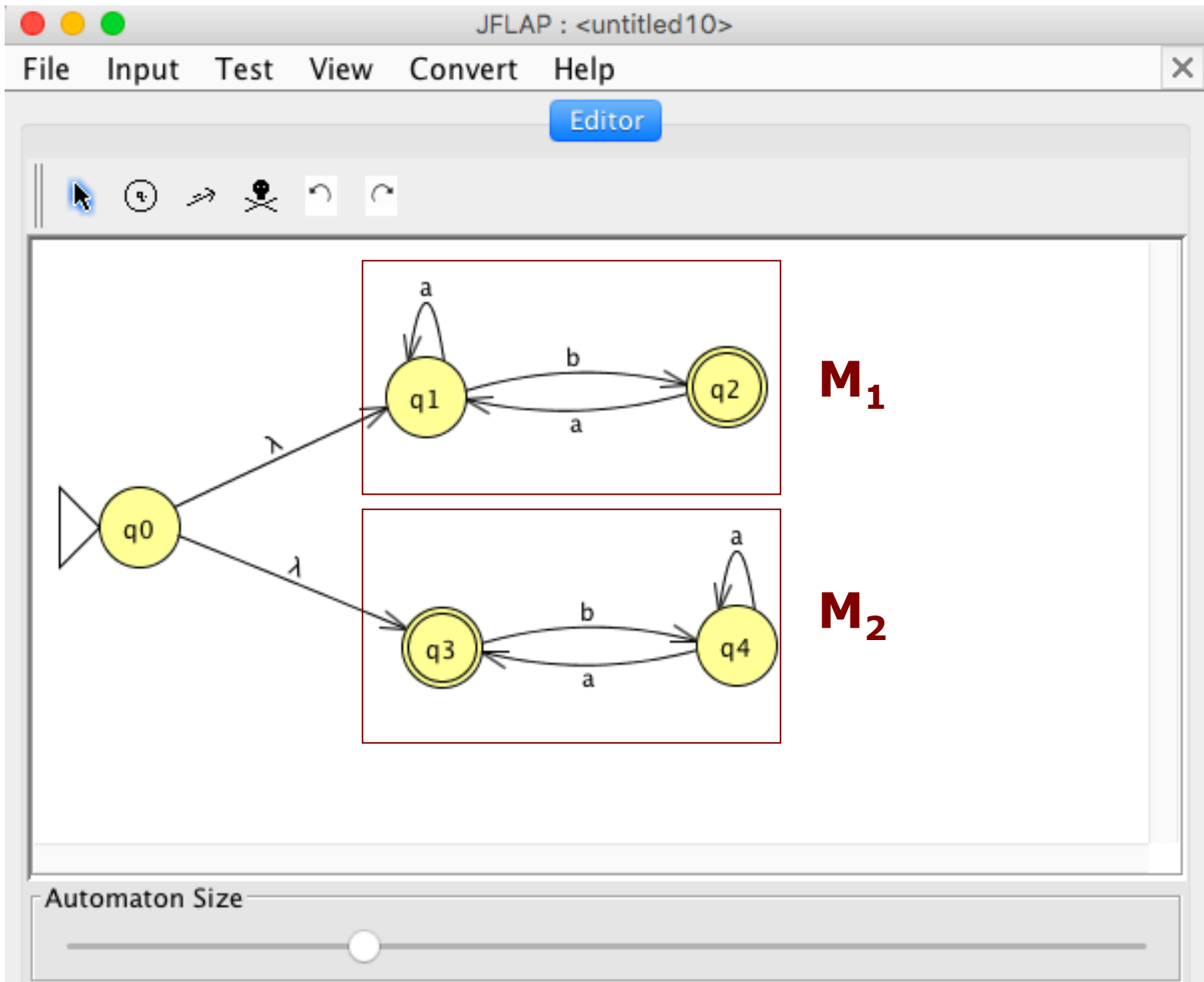
M₁

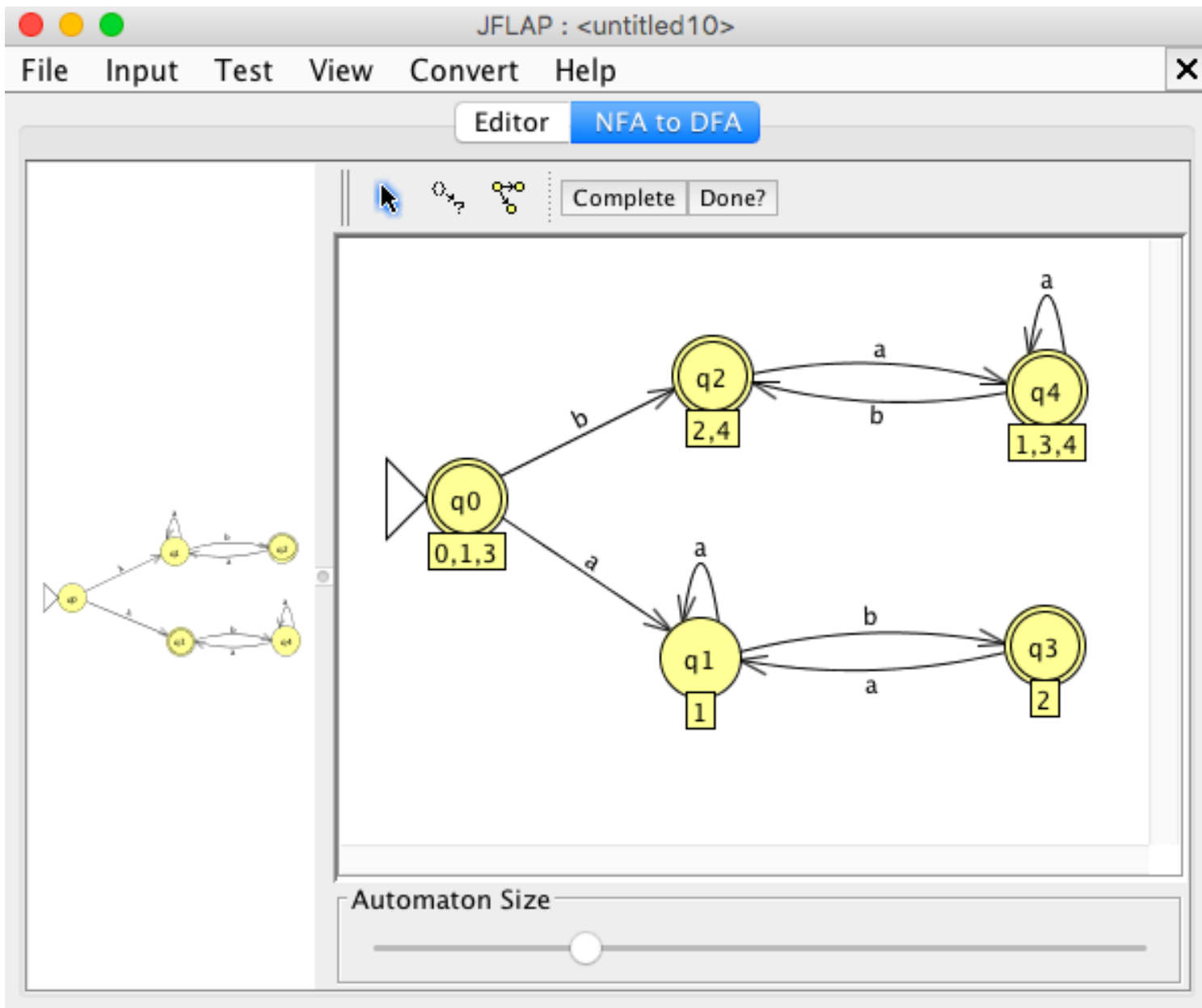


M₂









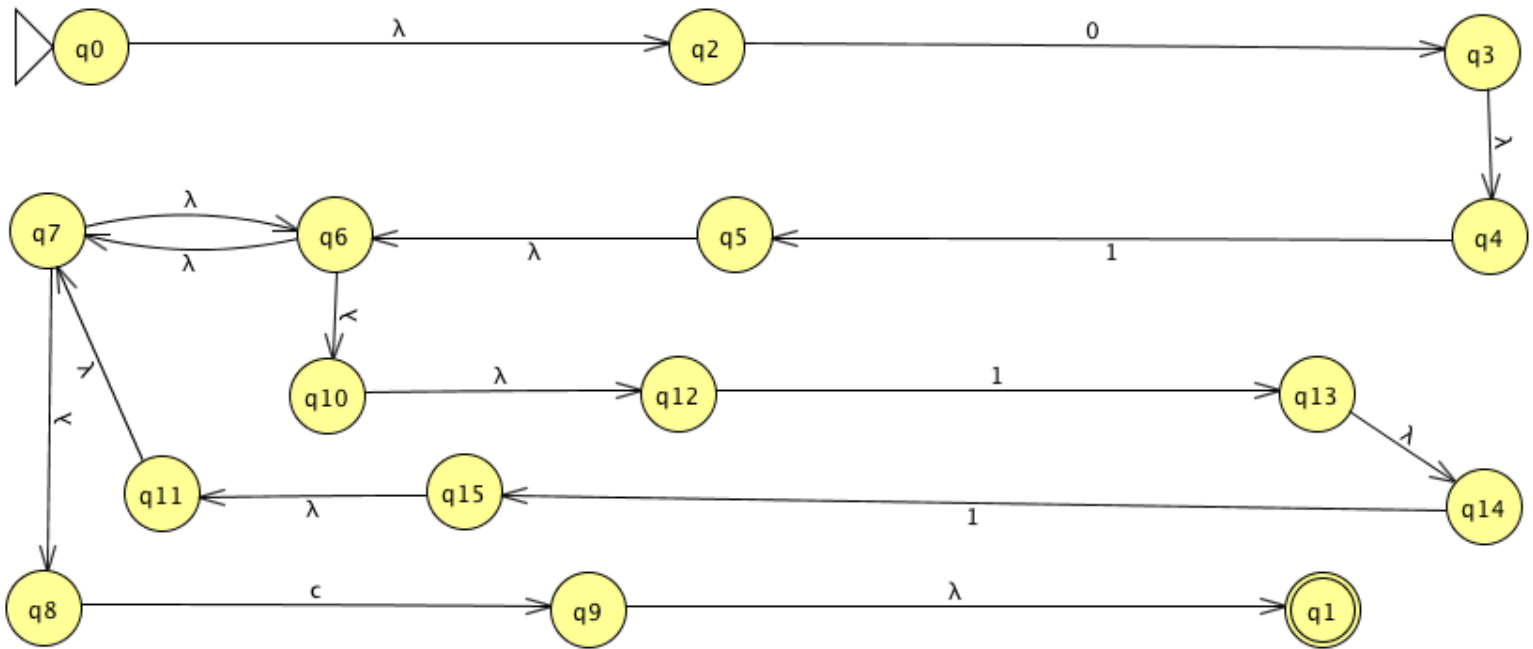
Esercizio 5: Sia L il linguaggio denotato dalla espressione regolare

$$01(11)^*c$$

1. Trovare un automa a stati finiti che riconosce L
2. Trasformare l'automa non deterministico in automa deterministico equivalente

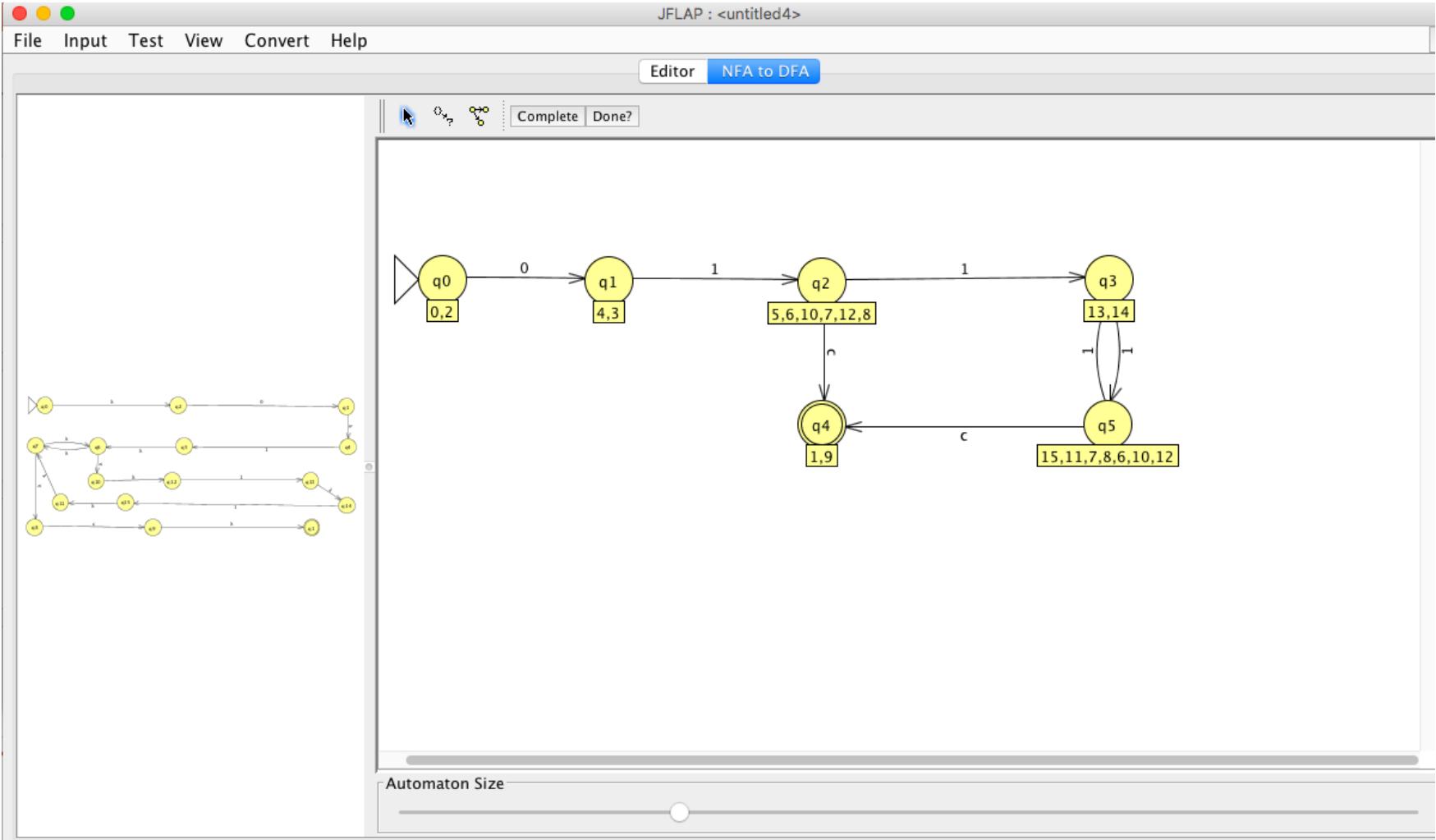
The automaton is complete.
 "Export" will put it in a new window.

Do Step Do All Export

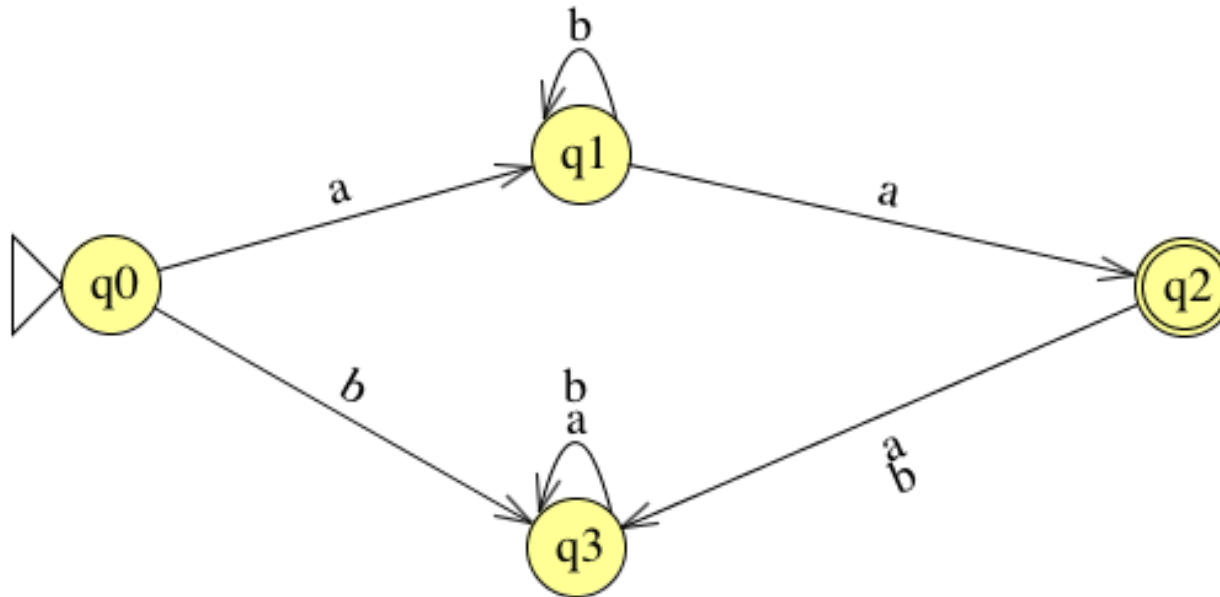


Automaton Size





Esercizio 6: Determinare il linguaggio riconosciuto dal seguente automa e scrivere la corrispondente espressione regolare



JFLAP : <untitled8>

File Input Test View Convert Help

Editor Convert to Grammar

Hint Show All What's Left? Export

LHS	RHS
B	λ
B	bC
C	aC
A	bA
C	bC
B	aC
A	aB
S	aA
S	bC

Table Text Size

JFLAP : <untitled8>

File Input Test View Convert Help

Editor Convert FA to RE

Generalized Transition Graph Finished!
ab*a

Do It Export

```
graph LR; q0((q0)) -- ∅ --> q0; q2(((q2))) -- ∅ --> q2; q0 -- ab*a --> q2; q2 -- ∅ --> q0; start(( )) --> q0; q2 --> end((( )))
```

Automaton Size

Slider control

JFLAP : <untitled10>

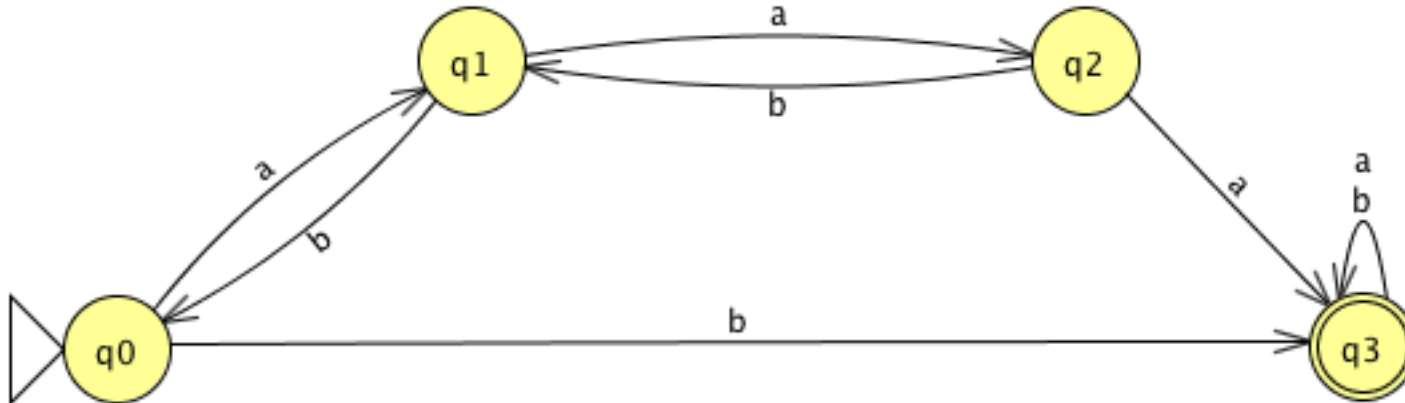
File Convert Help

Editor

Edit the regular expression below:

ab*a

Esercizio 7: Dato l'automa



1. Determinare il linguaggio riconosciuto dall'automa
2. Descrivere il linguaggio attraverso una espressione regolare

JFLAP : <untitled11>

File Input Test View Convert Help

Editor Convert to Grammar

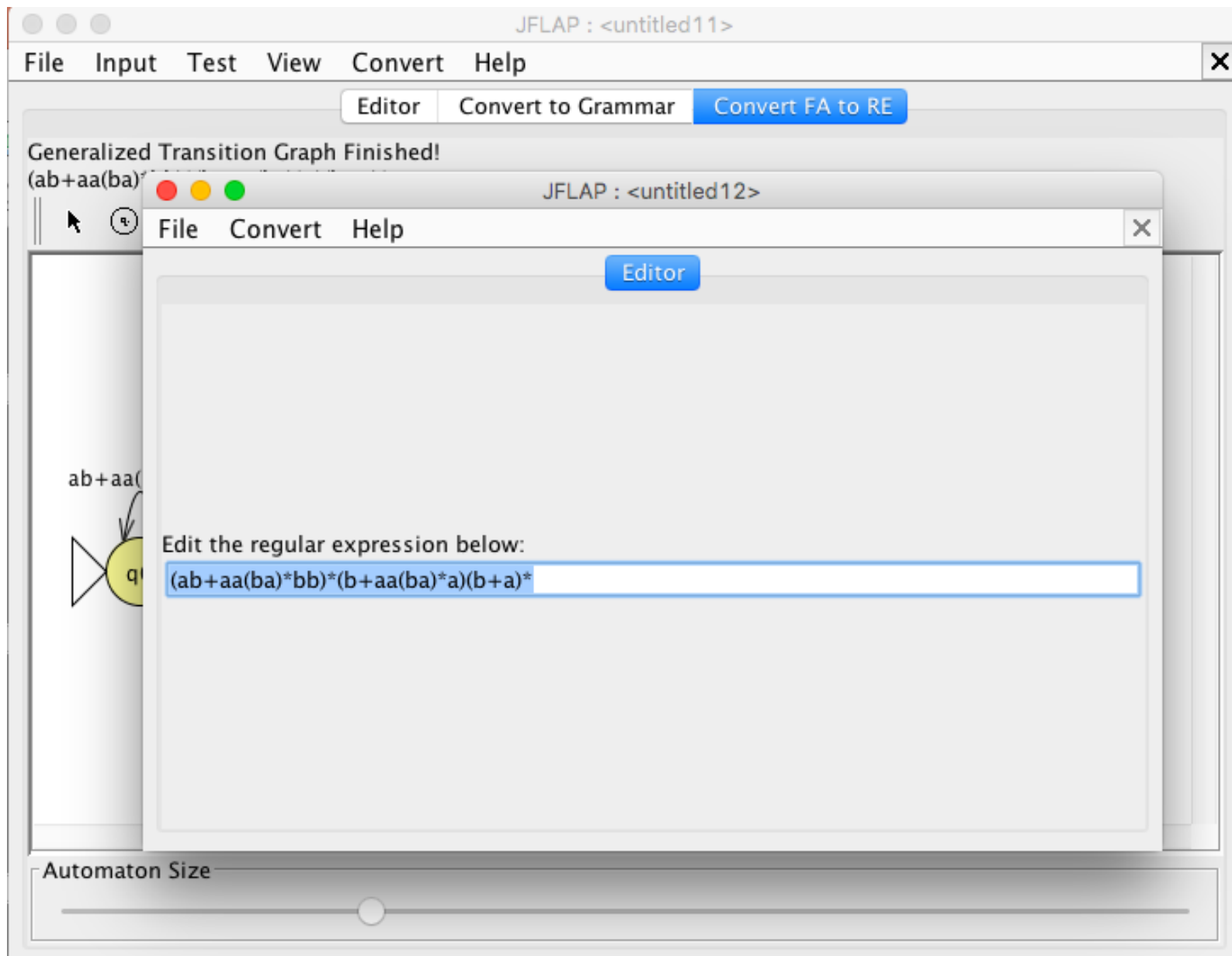
Hint Show All What's Left? Export

```

graph LR
    q0((q0)) -- a --> q1((q1))
    q1 -- b --> q0
    q1 -- a --> q2((q2))
    q2 -- b --> q1
    q2 -- a --> q3((q3))
    q3 -- b --> q2
    q0 -- b --> q3
  
```

LHS	RHS
C	→ aC
C	→ bC
S	→ bC
B	→ aC
S	→ aA
C	→ λ
A	→ bS
A	→ aB
B	→ bA

Table Text Size



Esercizio 8: dato il linguaggio

$L = \{w \text{ in } \{a,b\}^* : w \text{ contiene almeno tre 'b'}\}$

trovare un'espressione regolare per L

The screenshot shows the JFLAP interface with a DFA diagram and a table of test cases. The DFA has four states: q0 (start), q1, q2, and q3 (final). Transitions are: q0 to q0 on 'a', q0 to q1 on 'b'; q1 to q1 on 'a', q1 to q2 on 'b'; q2 to q2 on 'a', q2 to q3 on 'b'; q3 to q3 on 'a' and 'b'.

Input	Result
bbb	Accept
abbba	Accept
aabbb	Accept
abb	Reject
abbabbb	Accept
abbab	Accept
aaa	Reject
aabb	Reject
bbbbaaab	Accept

