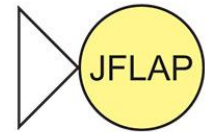


Una breve introduzione



- Software free per lo studio (anche) di
 - AUTOMI A STATI FINITI
 - LINGUAGGI FORMALI
- An interactive package for formal languages and automata

- Originariamente sviluppato da Susan H. Rodger (anni '90) in C/C++ con il nome di FLAP
- **Formal Languages and Automata Theory**
- Funzionalità FLAP relative allo studio di
 - Automi a stati finiti
 - Automi a pila
 - Macchine di Turing

- Dal 1994 il software è stato convertito in Java (**JFLAP**)
- Funzionalità JFLAP
 - Conversione da NFA in DFA, in grammatica regolare o in espr. regolare
 - Creazione di automi a pila a partire da grammatiche context-free
 - Studio dei parser LR e SLR

www.jflap.org

JFLAP

- [HOME](#)
- [What is JFLAP](#)
- [Get JFLAP](#)
- [JFLAP Tutorial](#)
(partially updated for
JFLAP 7.0)

Get JFLAP

INFORMATION about JFLAP:



- Get JFLAP Software

Please [fill out this form](#) and you can have the most recent version of JFLAP to use for free.

www2.cs.duke.edu

Web of Scienc...ist - Search UNIBA PhD site Webmail UNIBA Sensitrust Wilcoxon Sig...Data Science


JFLAP JFLAP SOFTWARE



[Home](#)
[Department](#)
[People](#)
[Research](#)
[Education](#)
[News & Media](#)
[Events](#)
[Industry Partners](#)
[Alumni](#)

JFLAP

Please fill out the form so we can track the usage of JFLAP. Your name and email can be left blank if you prefer.

Name (optional)
Email (optional)
Country 
University

Type of user? (check all that apply)

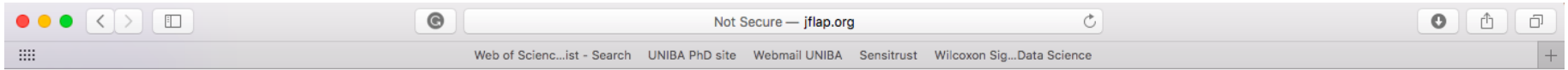
faculty graduate student undergraduate highschool teacher K-8 teacher K-12 student other

Which best describes how you plan to use JFLAP?

course I am teaching course I am taking in research other

If you are using JFLAP with a course, is JFLAP required to use?

YES NO



JFLAP SOFTWARE

NOTE: These are .jar files. If your operating system saves them as .zip files, rename them to .jar files. Then you should be able to click on them to run them.

NOTE 2: If you have trouble with clicking on the .jar file, try [jarfix](#)

NOTE 3: Most people will just want the software, if you want the source, scroll down to the bottom. NOTE 4: JFLAP Beta Version 8 was never finished. Recommend you use JFLAP Version 7.1.

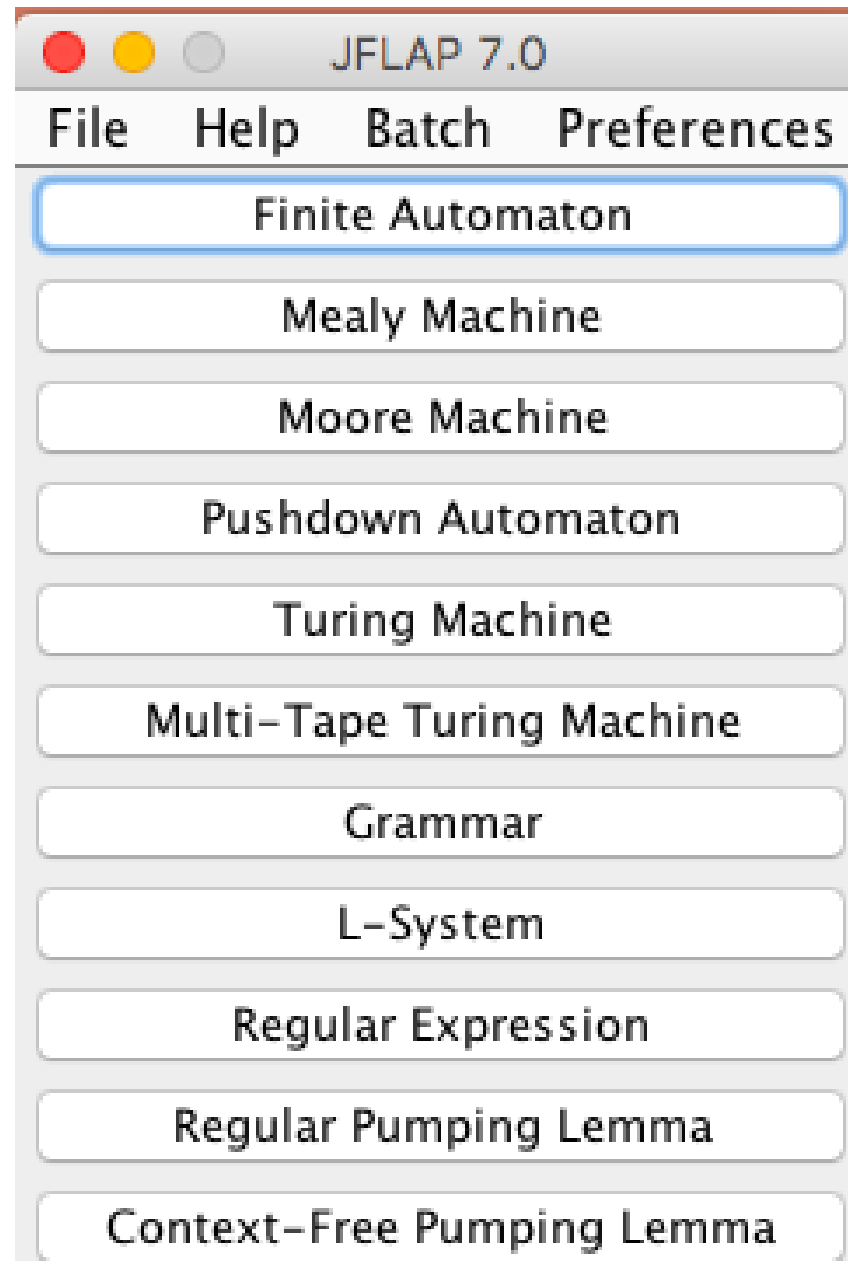
To get JFLAP SOFTWARE (jar file):

- JFLAP Software - will put newest version here as updates occur

This is a .jar file.

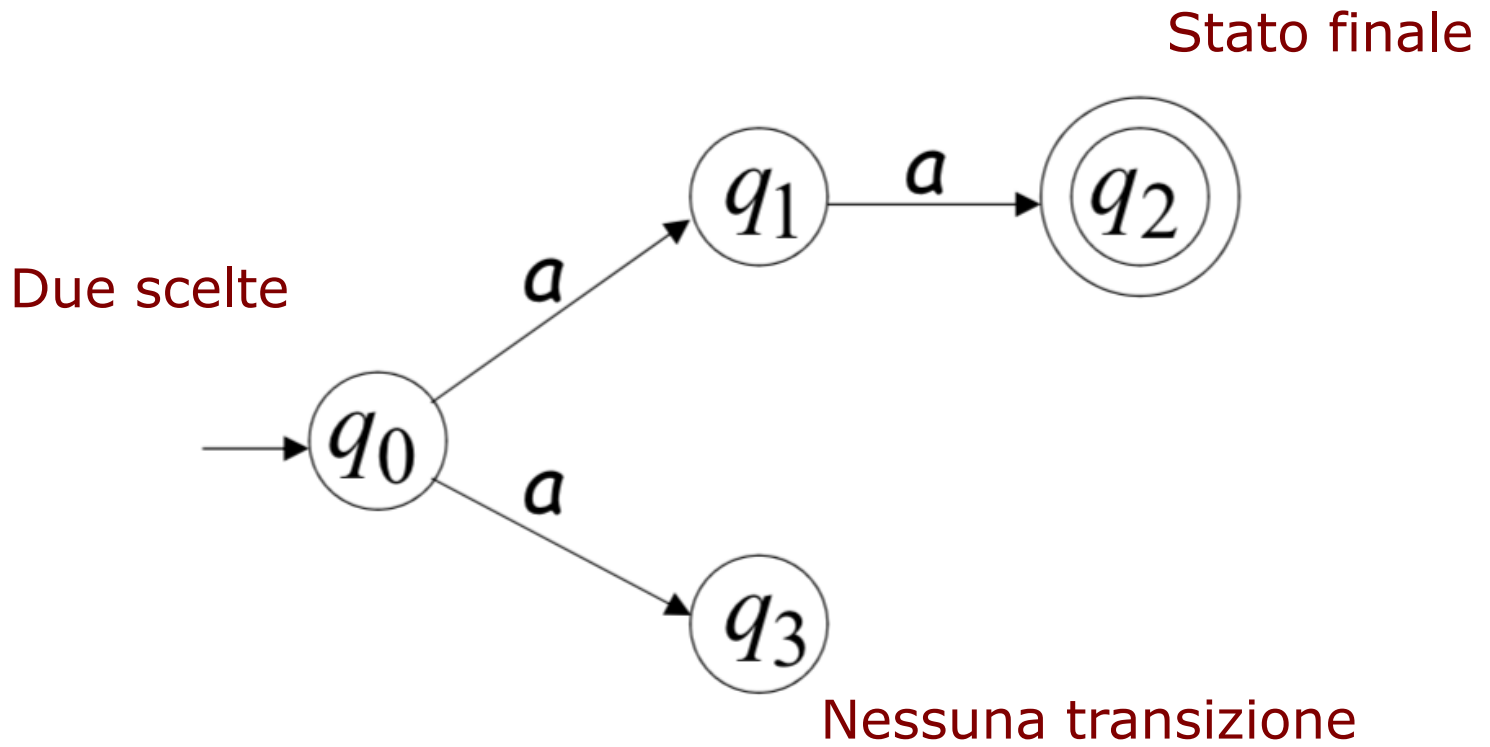
- Jul 27, 2018 - JFLAP Version 7.1
 - Note: JFLAP now uses Java 8.0.
 - [Changes made to JFLAP 7.0](#)
 - [JFLAP7.1.jar](#)
- Jan 24, 2015 - JFLAP Version 8.0 BETA
 - WARNING: This version is not complete and not as stable as Version 7. It is being released so people can use the new features. Strongly recommend you use the July 27, 2018 Version 7.1 instead if you are using JFLAP for a course.
 - [New items in JFLAP 8.0 BETA](#)
 - [JFLAP8_beta.jar](#)
- May 15, 2011 - JFLAP Version 7.0 (with SVG fixed and thinner JFLAP - We now provide one version with SVG and one with the SVG split out into a second .jar - read the changes file below)
 - NOTE: JFLAP uses Java 1.6. Mac users note that their default Java may be Java 1.5.
 - [Changes made to JFLAP](#)
 - Thinner JFLAP with no SVG. Most people will probably want this one. We also provide the SVG as a separate jar.

Usate JFLAP 7.0 → versione stabile



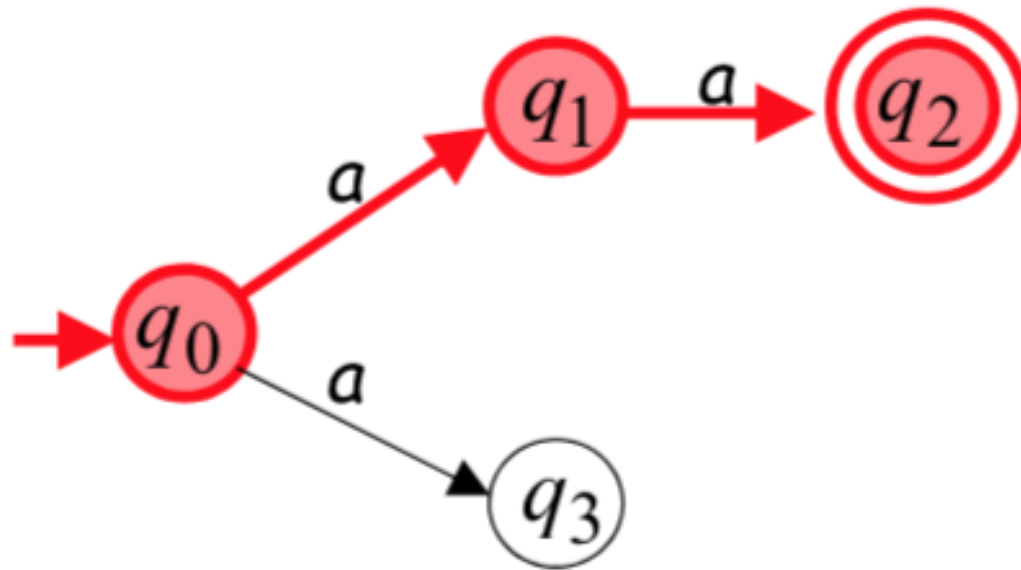
Esempio d'uso - Conversione NFA in DFA

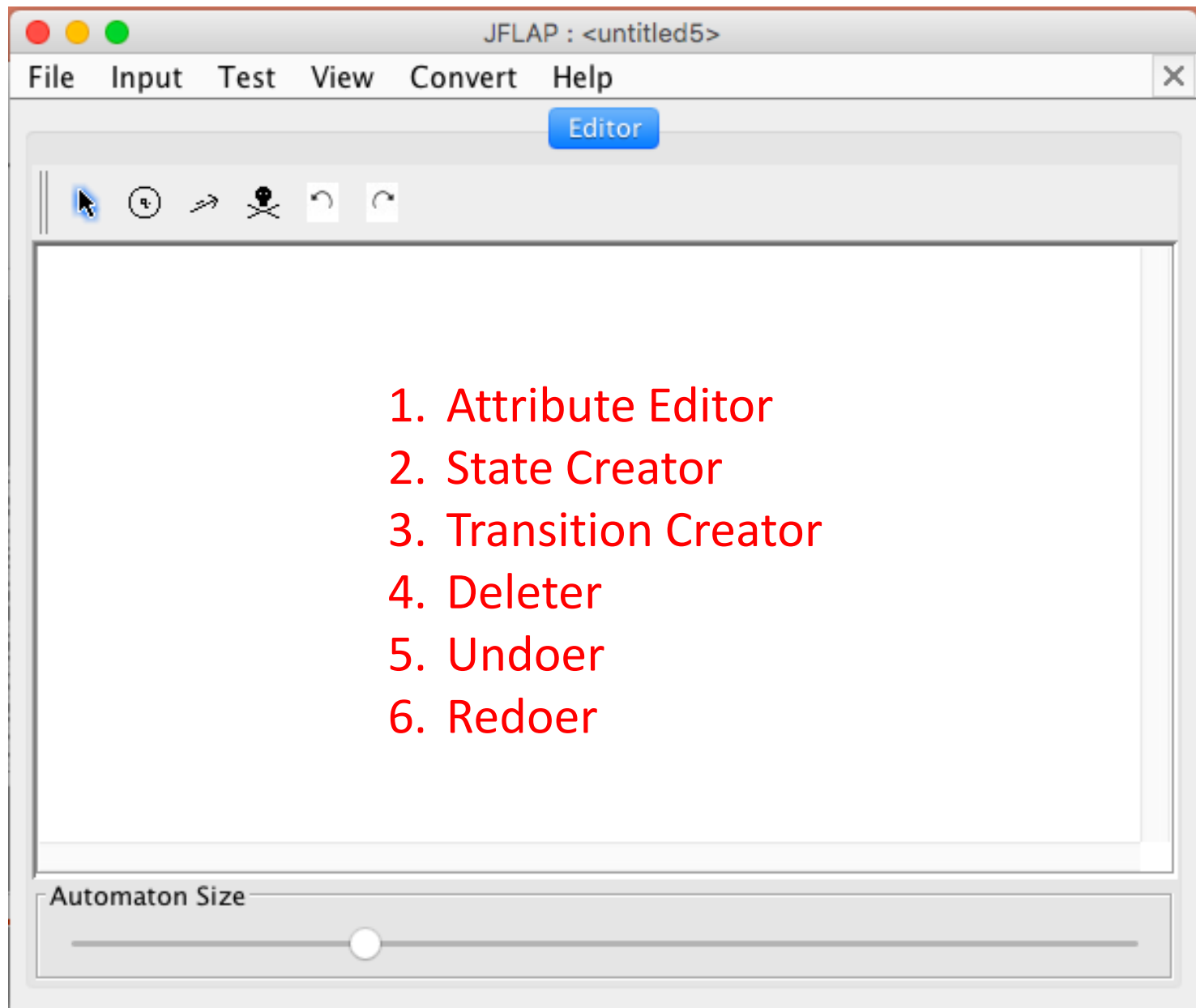
- Alfabeto di input = {a}

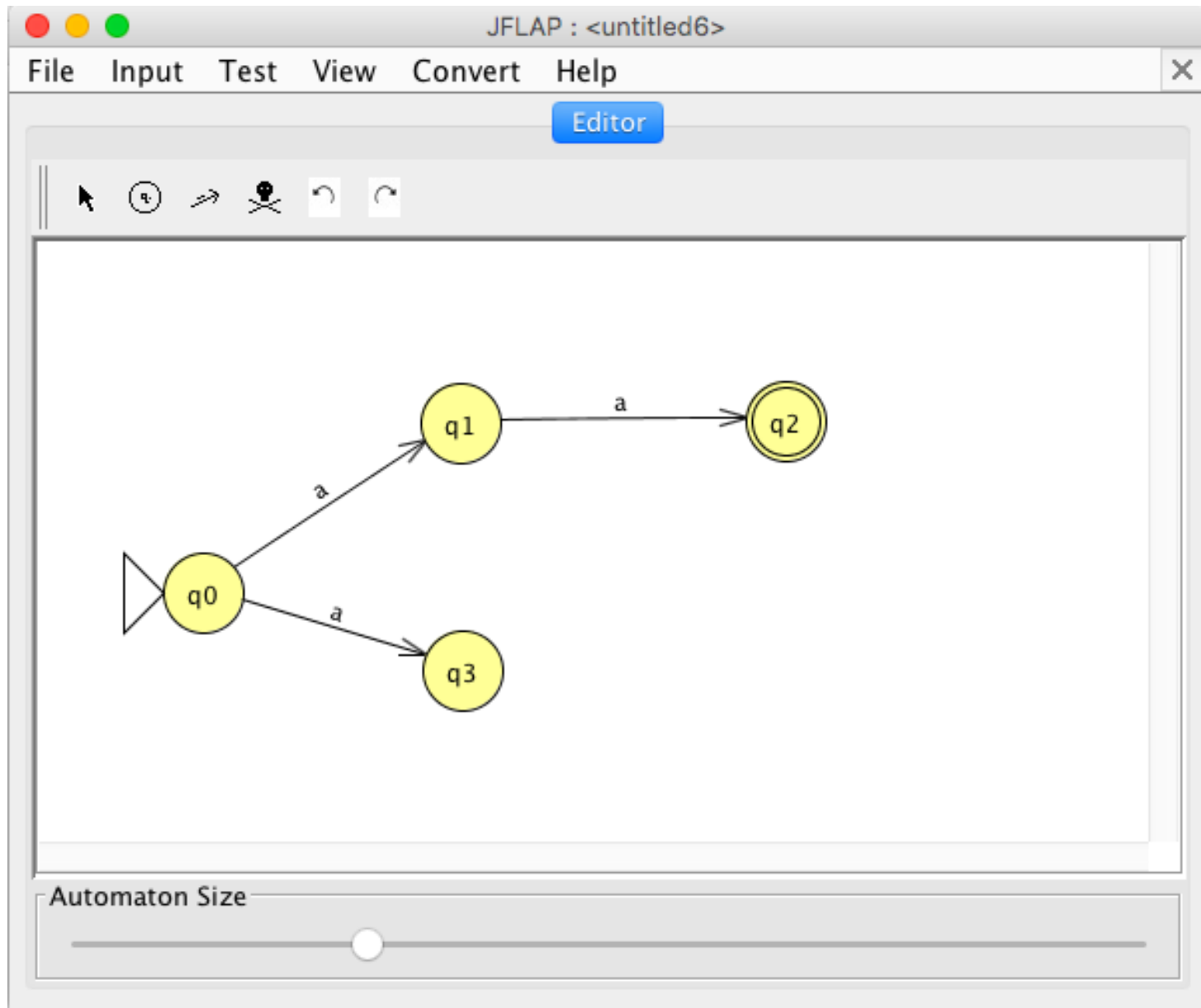


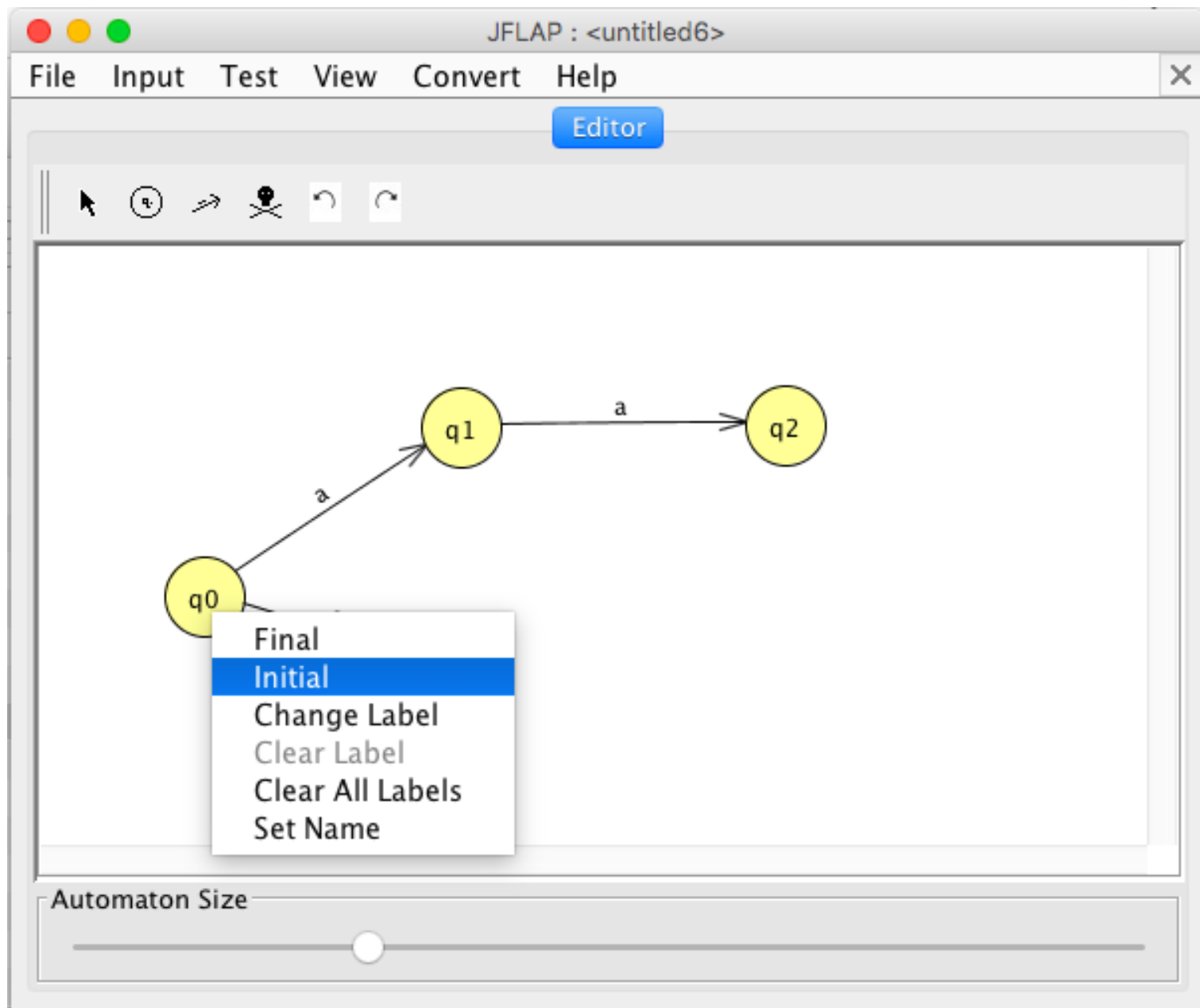
Un NFA **accetta** una stringa **se esiste** una computazione che **accetta la stringa**, cioè:

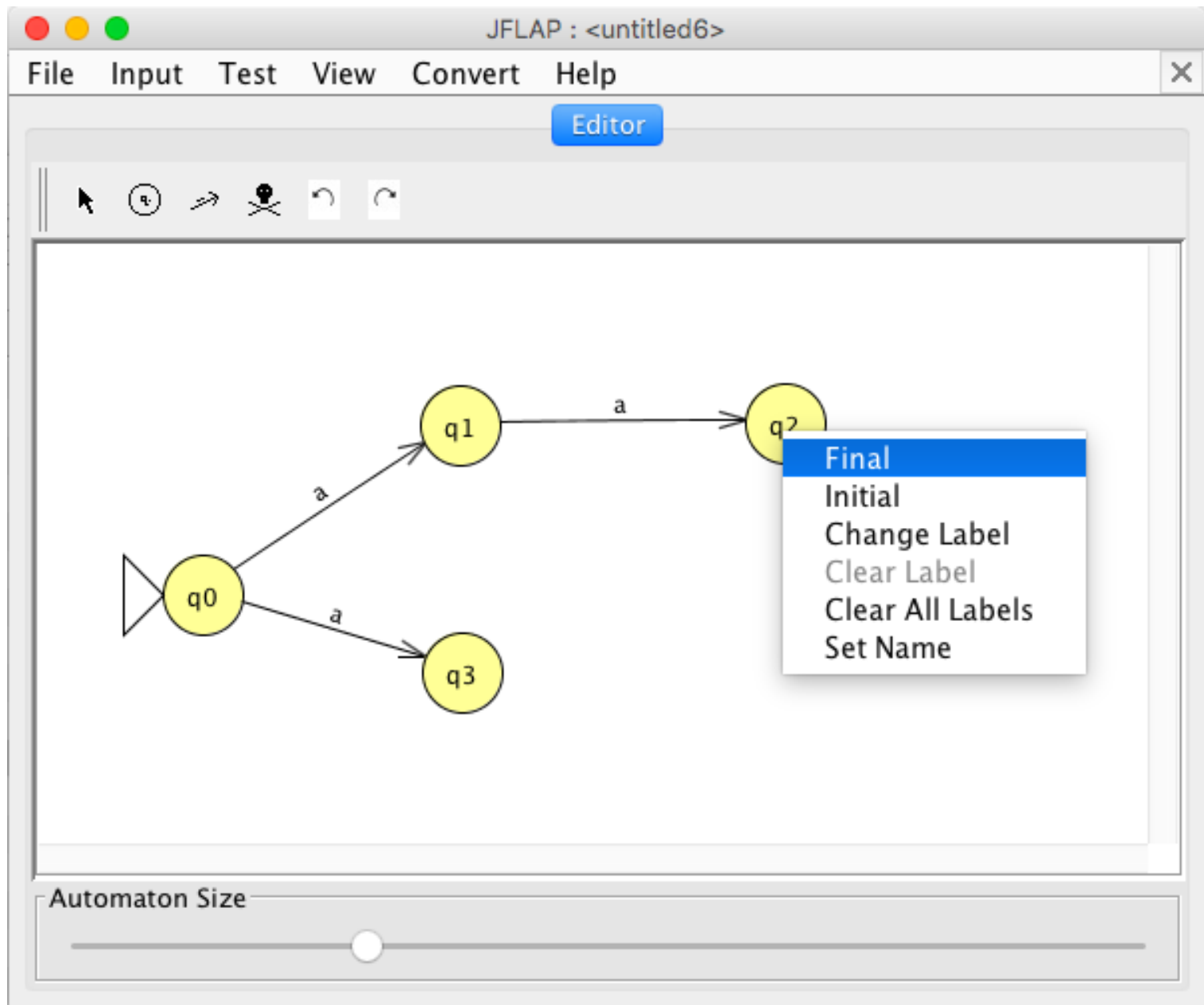
- tutta la stringa di input è accettata e
- l'automa si trova in uno stato finale











JFLAP : <untitled6>

File **Input** Test View Convert Help

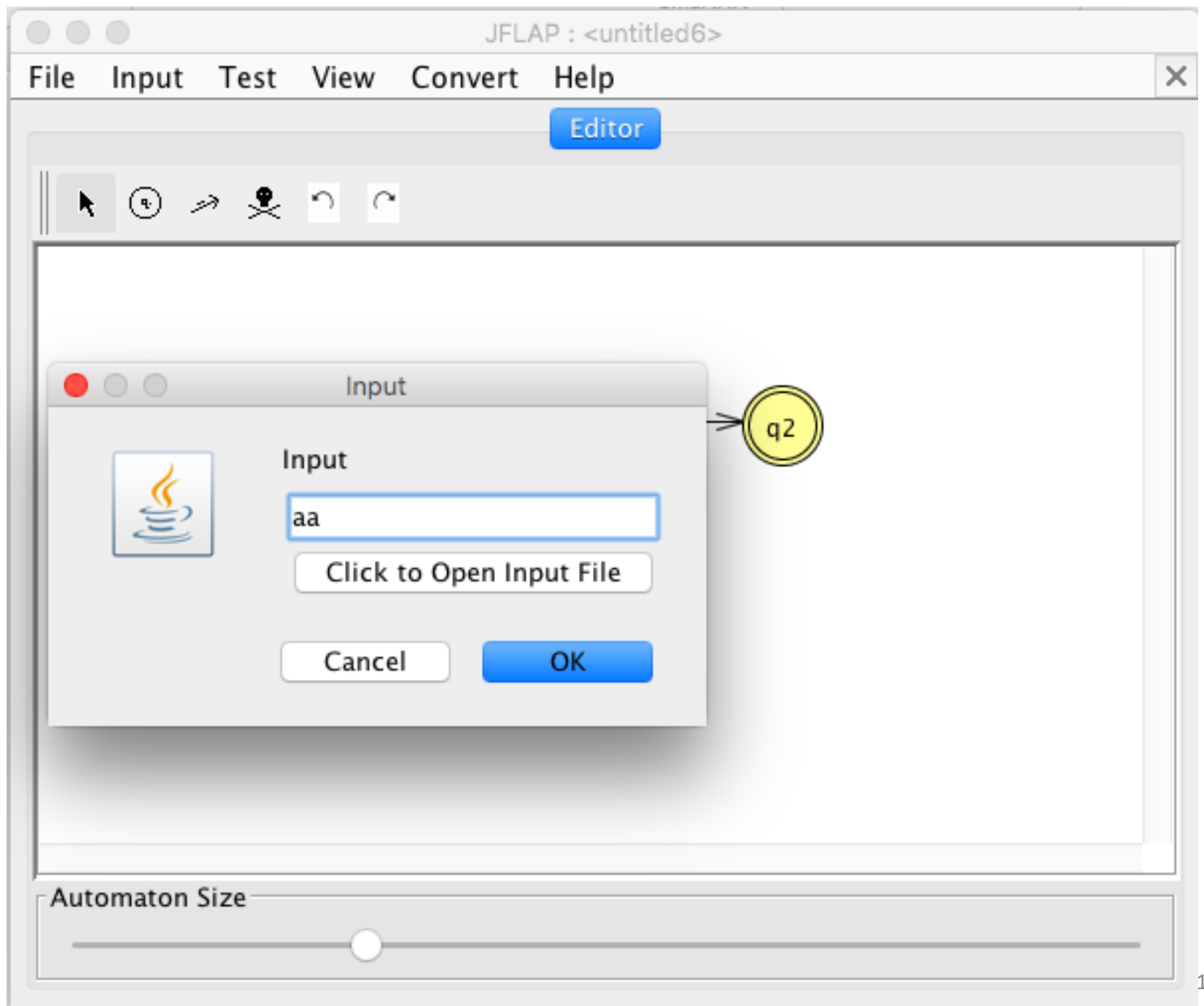
Step with Closure... ⌘R
Step by State... ⌘R
Fast Run...
Multiple Run ⌘M

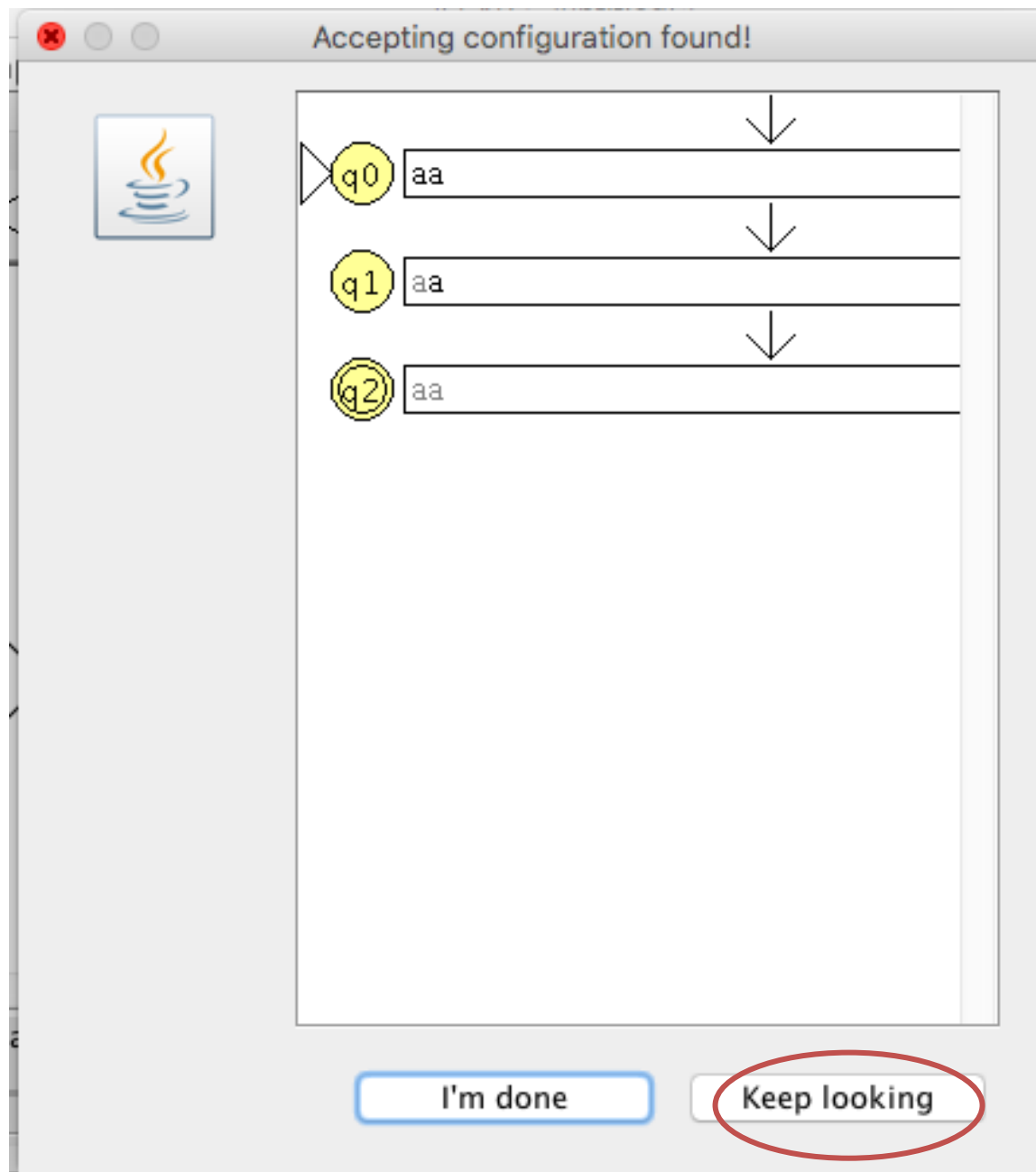
Editor

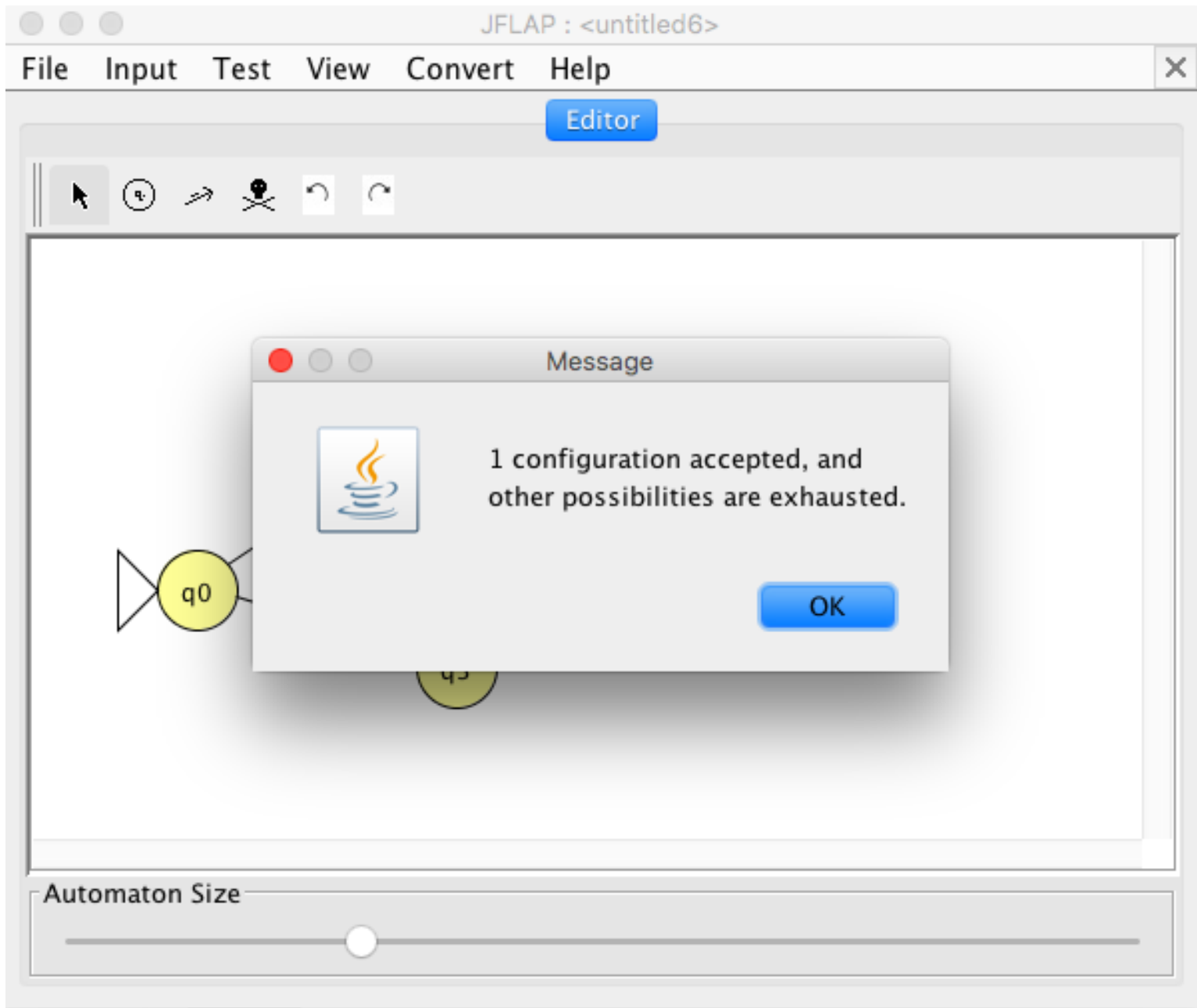
```
graph LR; q0((q0)) -- a --> q1((q1)); q0 -- a --> q3((q3)); q1 -- a --> q2(((q2)))
```

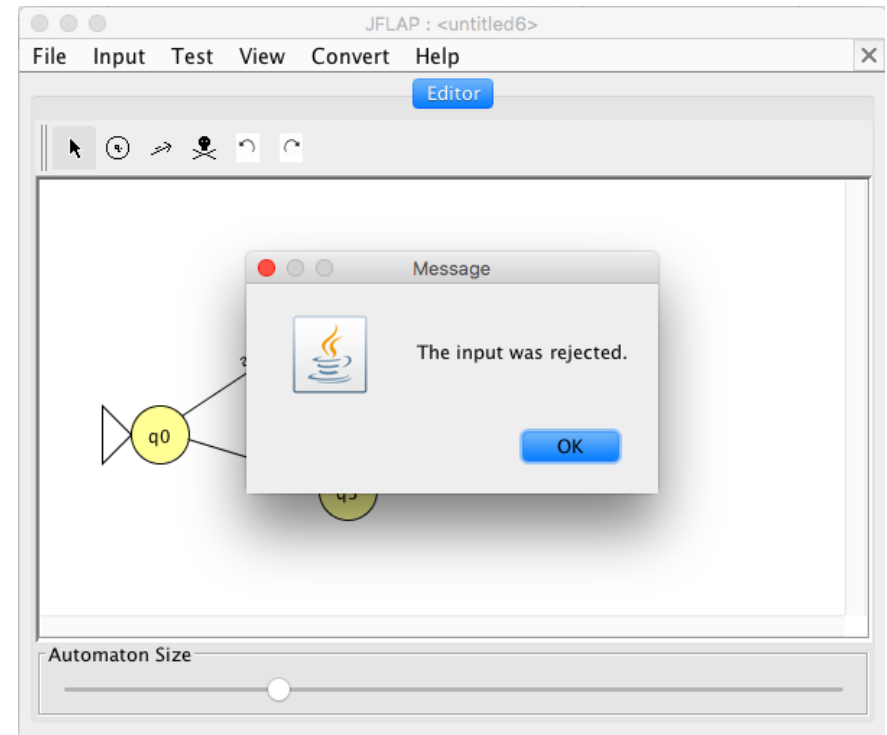
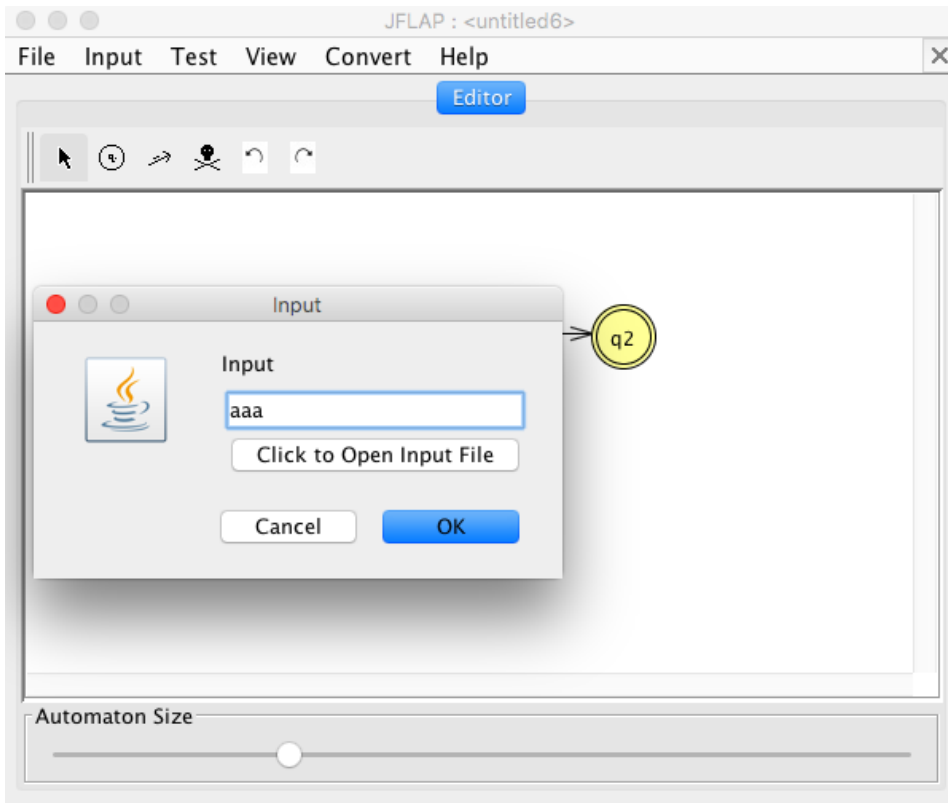
Automaton Size

Slider control for Automaton Size









JFLAP : <untitled6>

File **Input** Test View Convert Help

Step with Closure... ⌘R
Step by State... ⌘⇧R
Fast Run...
Multiple Run ⌘M

Editor

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

Automaton Size

Slider control for Automaton Size

JFLAP : <untitled6>

File Input Test View Convert Help

Editor Multiple Run

```

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

Table Text Size

Input	Result
aa	
a	
aaa	

Load Inputs Run Inputs Clear Enter Lambda

JFLAP : <untitled6>

File Input Test View Convert Help

Editor Multiple Run

```

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

Table Text Size

Input	Result
aa	Accept
a	Reject
aaa	Reject

Load Inputs Run Inputs Clear Enter Lambda

JFLAP : <untitled6>

File **Input** Test View Convert Help

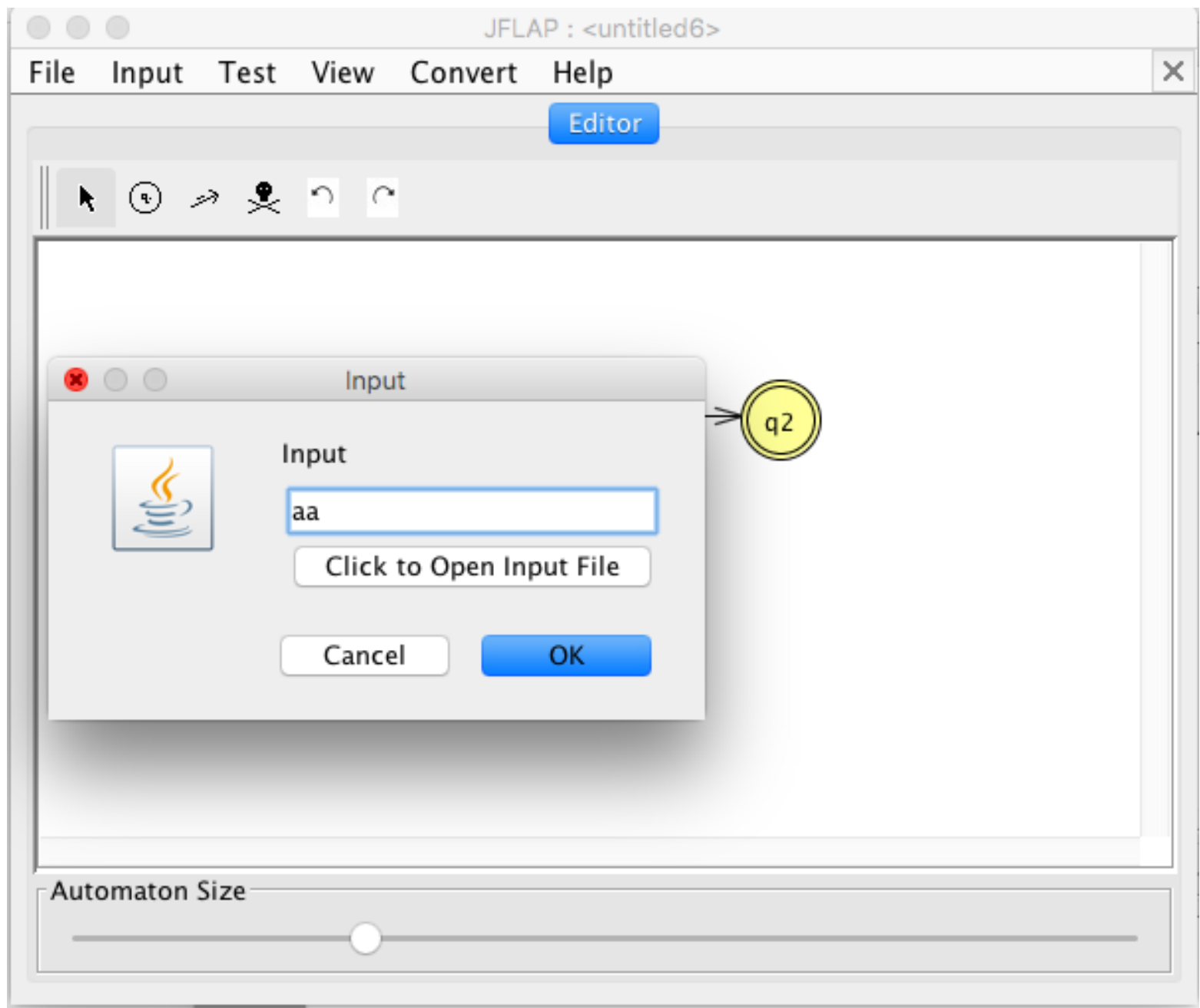
Step with Closure... ⌘R
Step by State... ⇧⌘R
Fast Run...
Multiple Run ⌘M

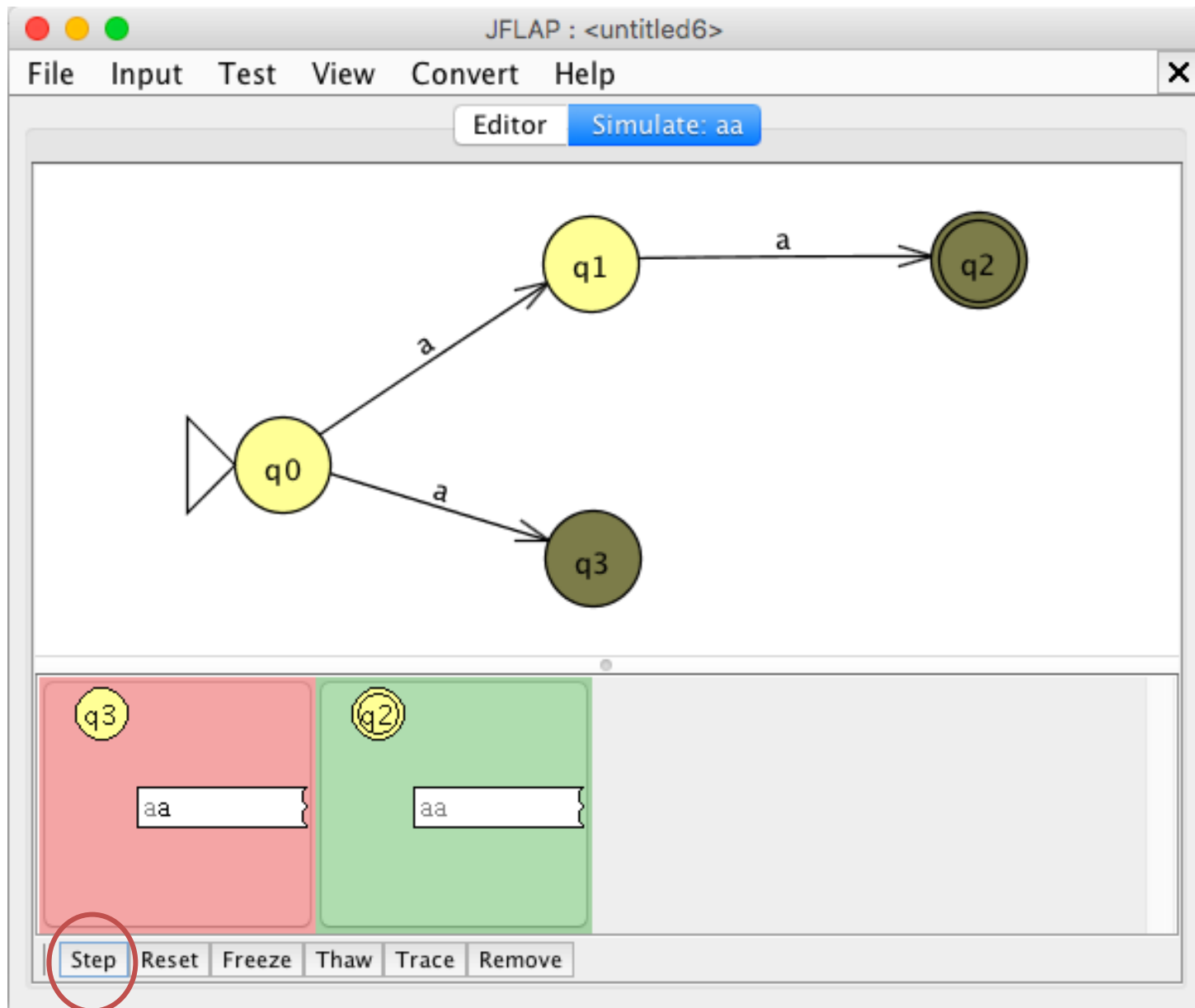
Editor

```
graph LR; start(( )) --> q0((q0)); q0 -- a --> q1((q1)); q0 -- a --> q3((q3)); q1 -- a --> q2(((q2))); style start fill:none,stroke:none; style q2 stroke-width:4px;
```

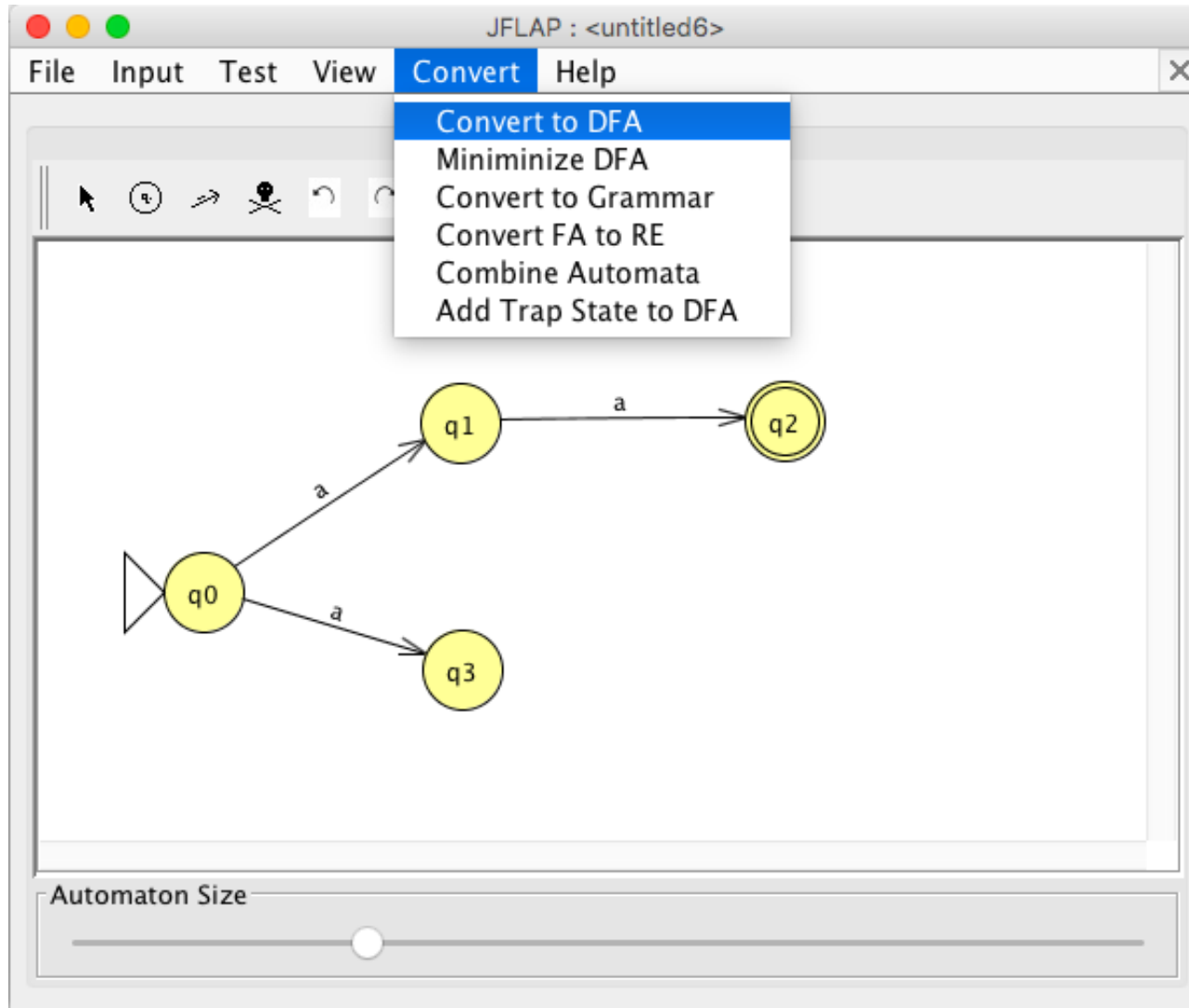
Automaton Size

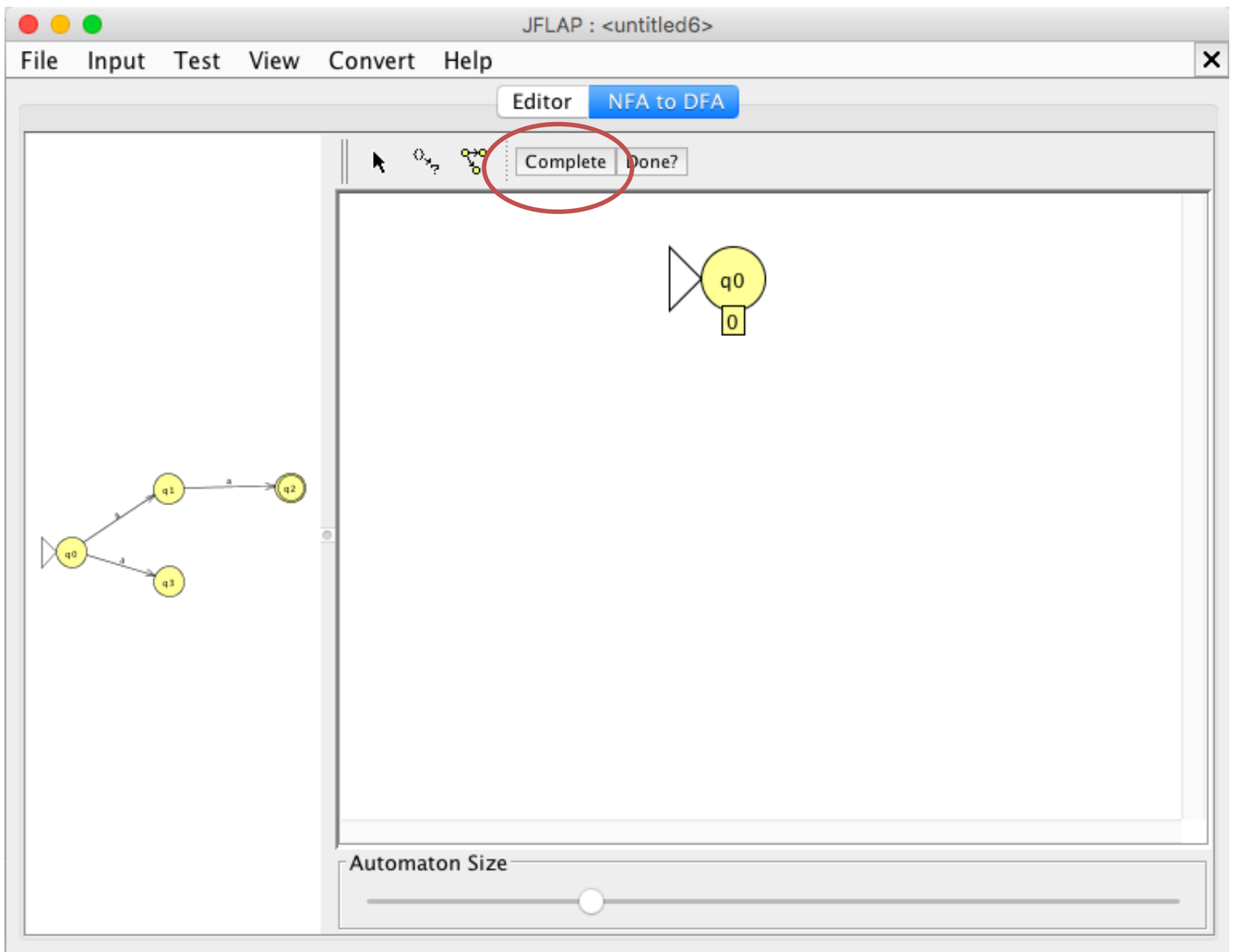
Slider control for Automaton Size





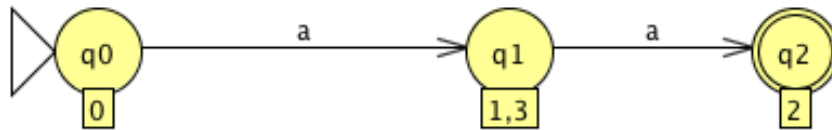
Conversione NFA in DFA



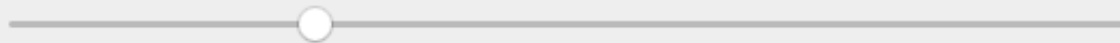


Editor NFA to DFA

Complete Done?



Automaton Size



JFLAP : <untitled6>

File Input Test View **Convert** Help

- Convert to DFA
- Minimize DFA
- Convert to Grammar**
- Convert FA to RE
- Combine Automata
- Add Trap State to DFA

Convert FA to DFA Done?

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

```
graph LR; q0((q0)) -- a --> q1((q1)); q1 -- a --> q2(((q2)))
```

Automaton Size

JFLAP : <untitled6>

File Input Test View Convert Help

Editor NFA to DFA Convert to Grammar

Hint Show All What's Left? Export

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

LHS	RHS
B	λ
S	aA
A	aB
S	aC

Table Text Size

JFLAP : <untitled8>

File Input **Test** Convert Help

Test for Grammar Type

Table Text Size

LHS	RHS
S	→ aA
S	→ aC
B	→ λ
A	→ aB

```

graph LR
    q0((q0)) -- S --> q0
    q0 -- A --> q1((q1))
    q0 -- C --> q3((q3))
    q1 -- A --> q2((q2))
    q3 -- B --> q2
  
```

Table Text Size

JFLAP : <untitled8>


File Input Test Convert Help

Editor

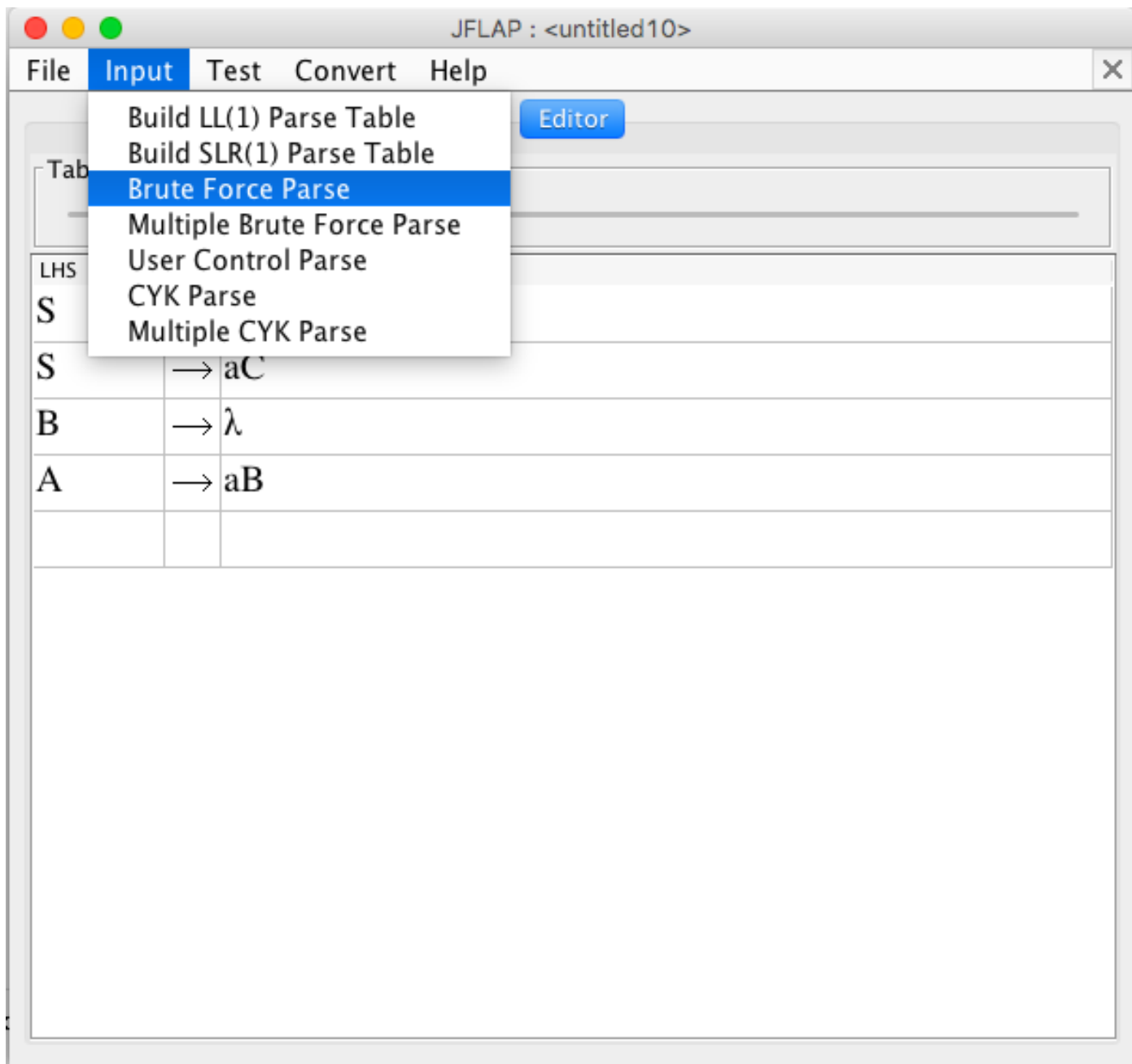
Table Text Size

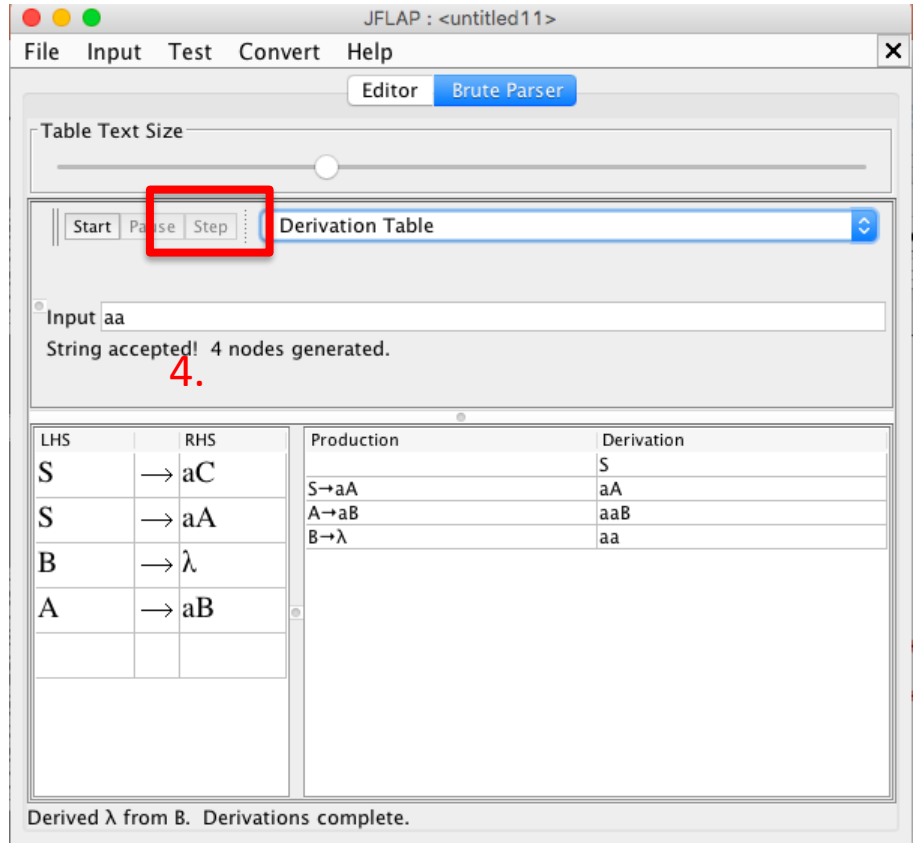
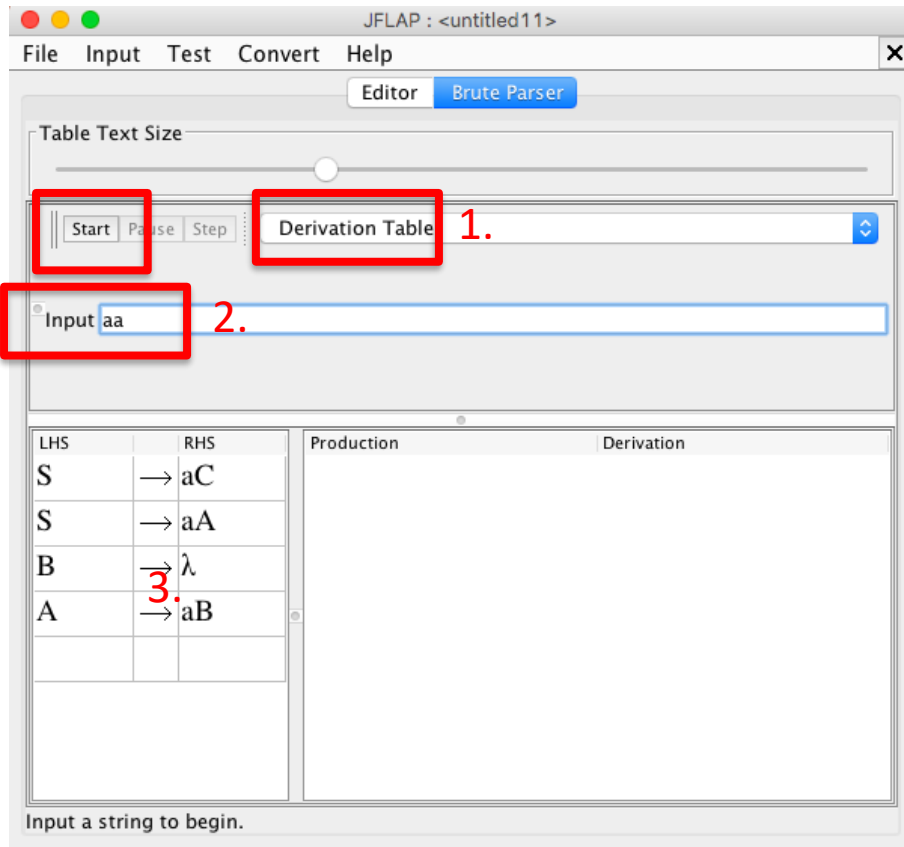
LHS	RHS
S	→ aA
S	→ aC
B	→ λ

Grammar Type

 This is a right-linear Grammar (Regular Grammar and Context-Free Grammar)

OK

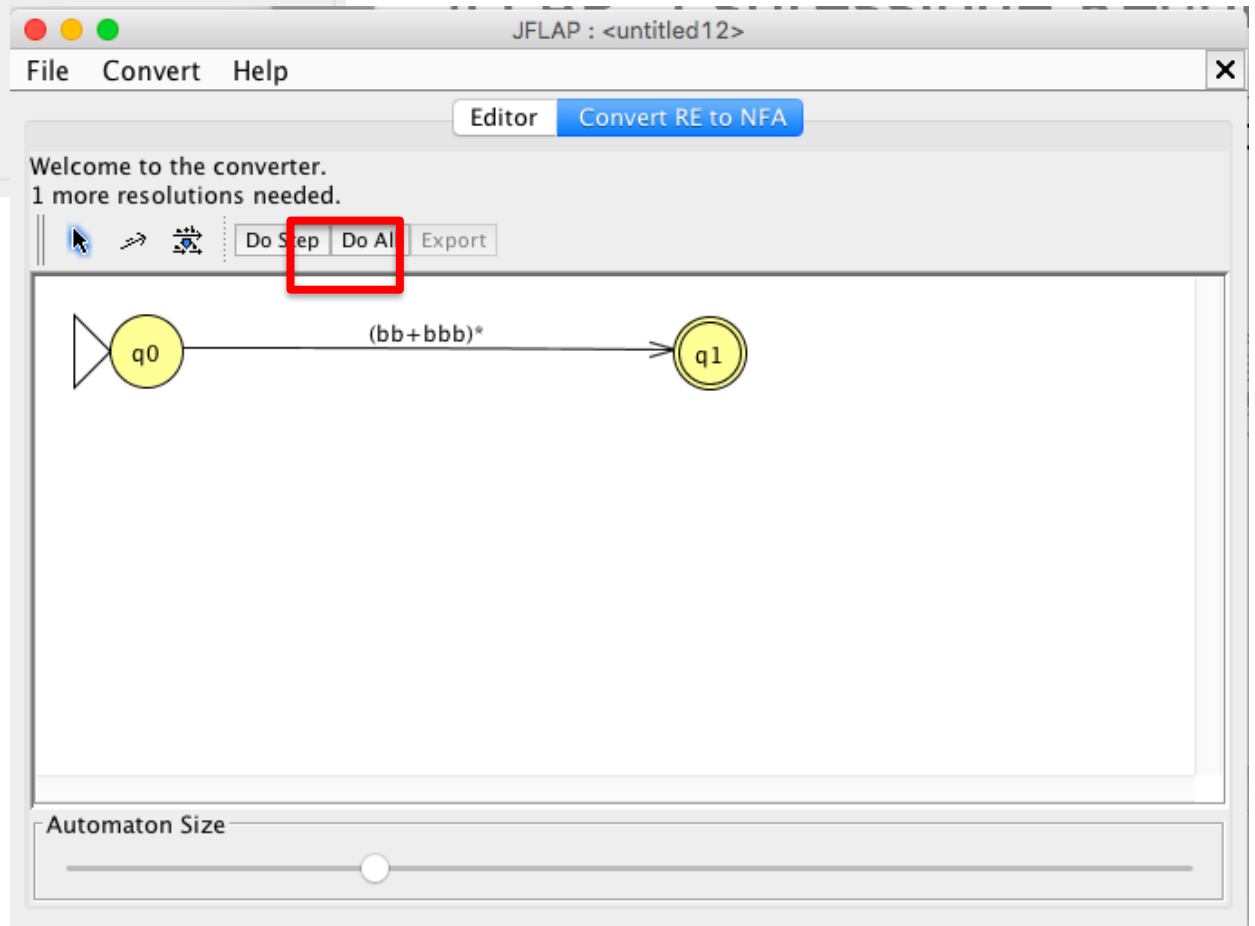
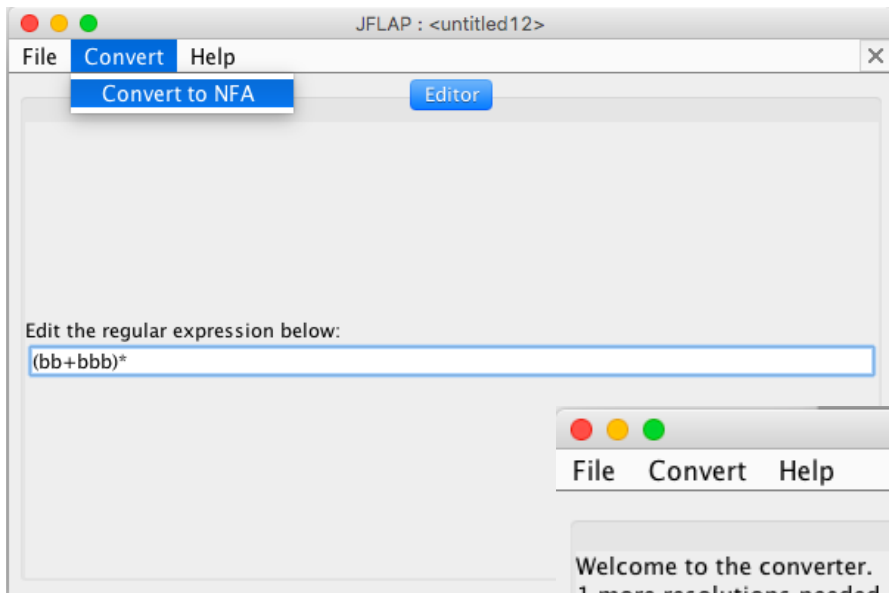




Esempio d'uso – Espressione regolare

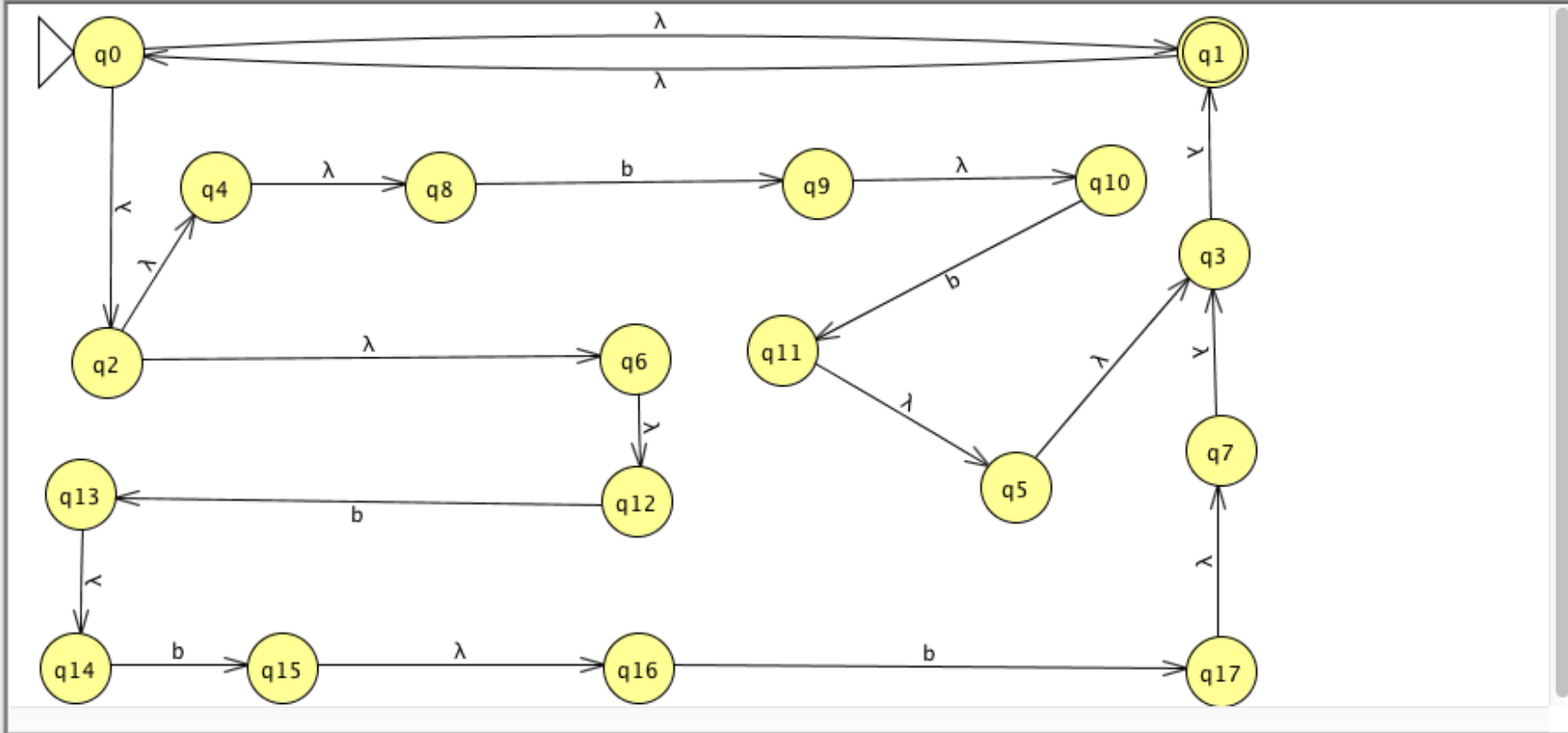
Sia L il linguaggio denotato dalla espr. regolare $(bb+bbb)^*$;

definire un automa a stati finiti che riconosce L ;
trasformare l'automa del punto 1 in DFA equivalente.

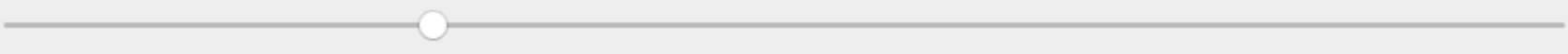


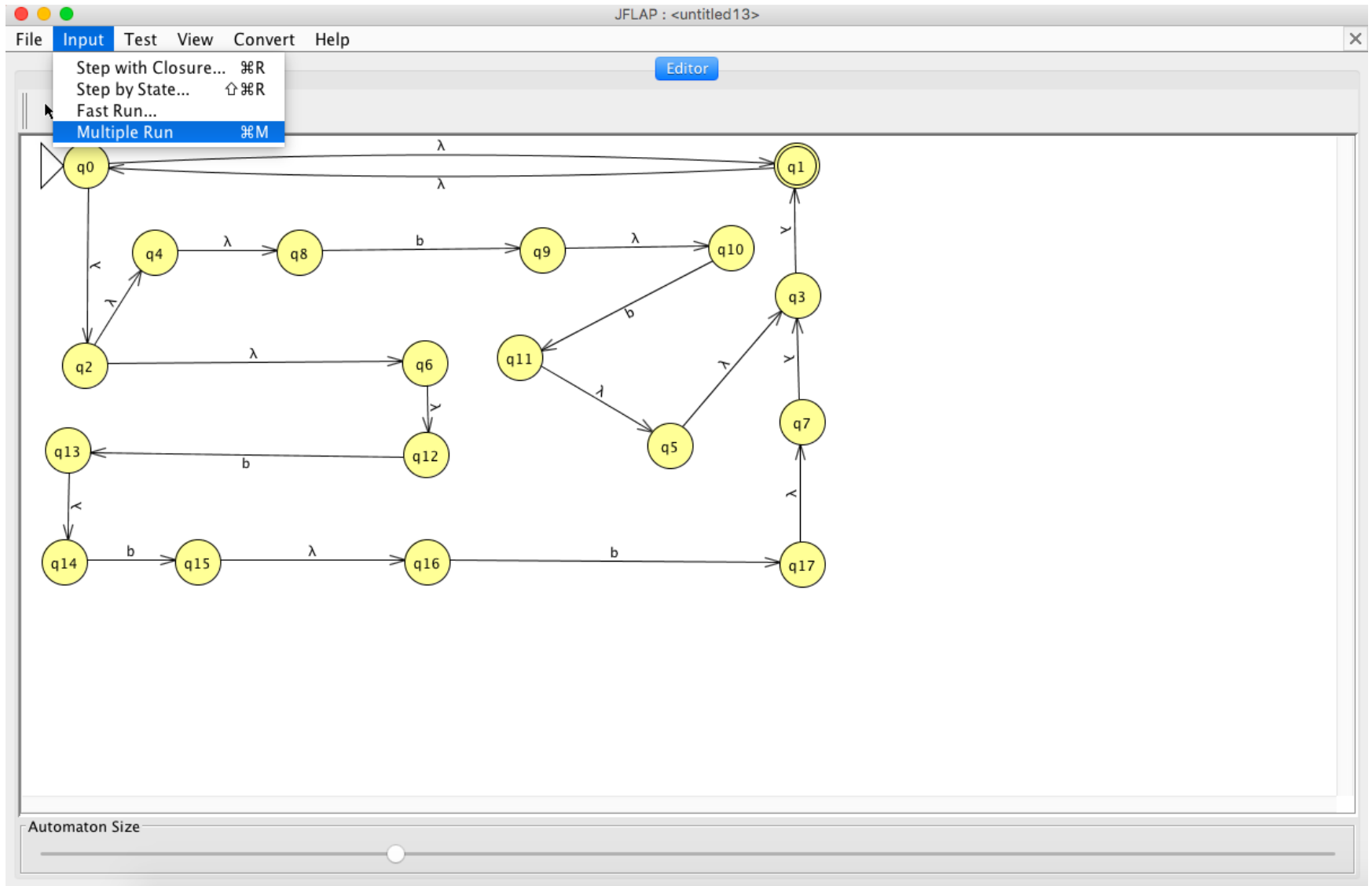
The automaton is complete.
 "Export" will put it in a new window.

Do Step Do All Export



Automaton Size





JFLAP : <untitled13>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
	Accept
b	Reject
bb	Accept
bbbb	Accept
bbb	Accept
bbbbb	Accept
bbbbbbb	Accept

Load Inputs Run Inputs Clear Enter Lambda View Trace

Esempio d'uso – Grammatiche regolari

Una grammatica regolare è una grammatica lineare destra o sinistra.

I linguaggi generati da una grammatica regolare sono tutti e soli i linguaggi regolari

Data la grammatica lineare destra
 $G=(X,V,S,P)$ con

$$- X = \{a, b\}$$

$$- V = \{S, B\}$$

$$- P = \{S \rightarrow aB, \\ B \rightarrow aB \mid bS \mid a\}$$


determinare un DFA M tale che $L(G) = L(M)$

JFLAP : <untitled14>

File Input Test Convert Help

Editor

Table Text Size



LHS		RHS
S	→	aB
B	→	aB
B	→	bS
B	→	a

JFLAP : <untitled14>

File Input Test **Convert** Help

Convert CFG to PDA (LL)
 Convert CFG to PDA (LR)
Convert Right-Linear Grammar to FA
 Transform Grammar

LHS	RHS
S	→ aB
B	→ aB
B	→ bS
B	→ a

Export

LHS	RHS
S	→ B(a)B
B	→ B(a)B
B(a)	→ a
B	→ B(b)S
B(b)	→ b
B	→ a

Export

JFLAP : <untitled14>

File Input Test Convert Help

Editor Chomsky Converter **Convert to FA**

Table Text Size

Production	Created
S → aB	<input checked="" type="checkbox"/>
B → aB	<input checked="" type="checkbox"/>
B → bS	<input checked="" type="checkbox"/>
B → a	<input checked="" type="checkbox"/>

Show All **Create Selected** Done? Export

Automaton Size

JFLAP : <untitled14>

File Input Test Convert Help

Editor Chomsky Converter Convert to FA

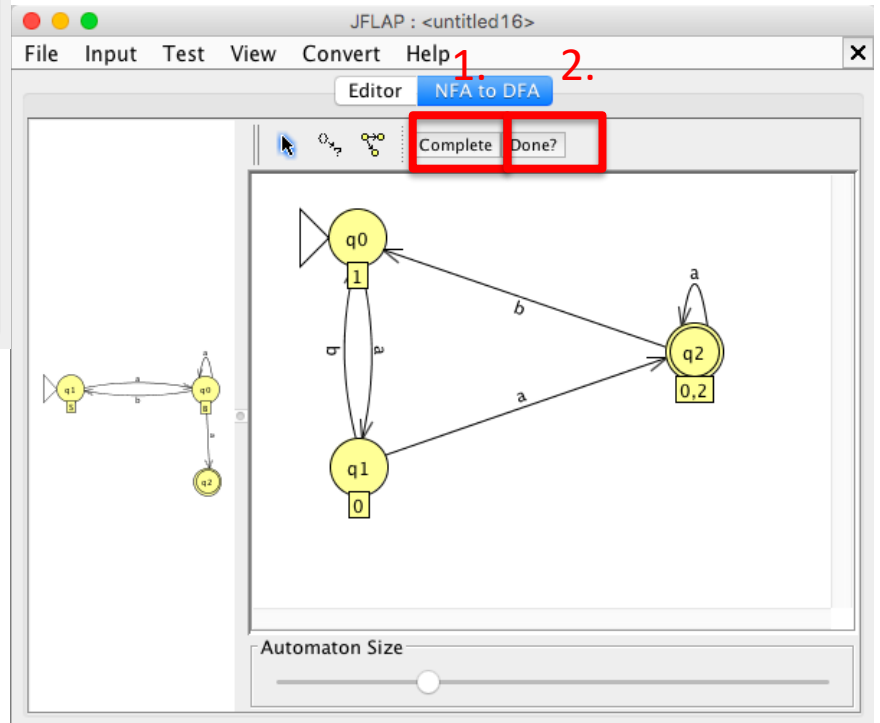
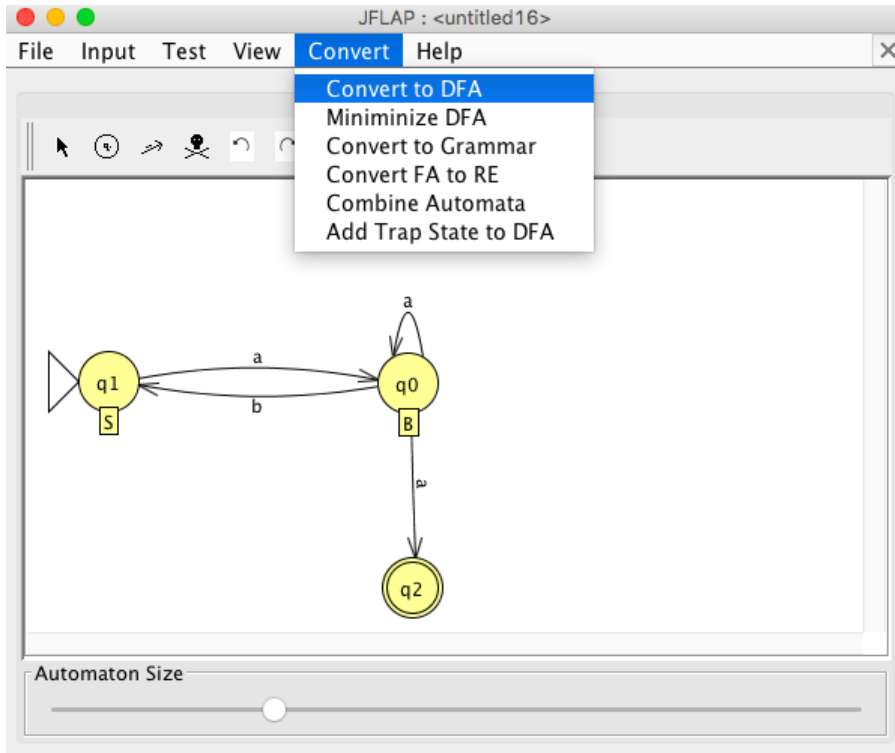
Table Text Size

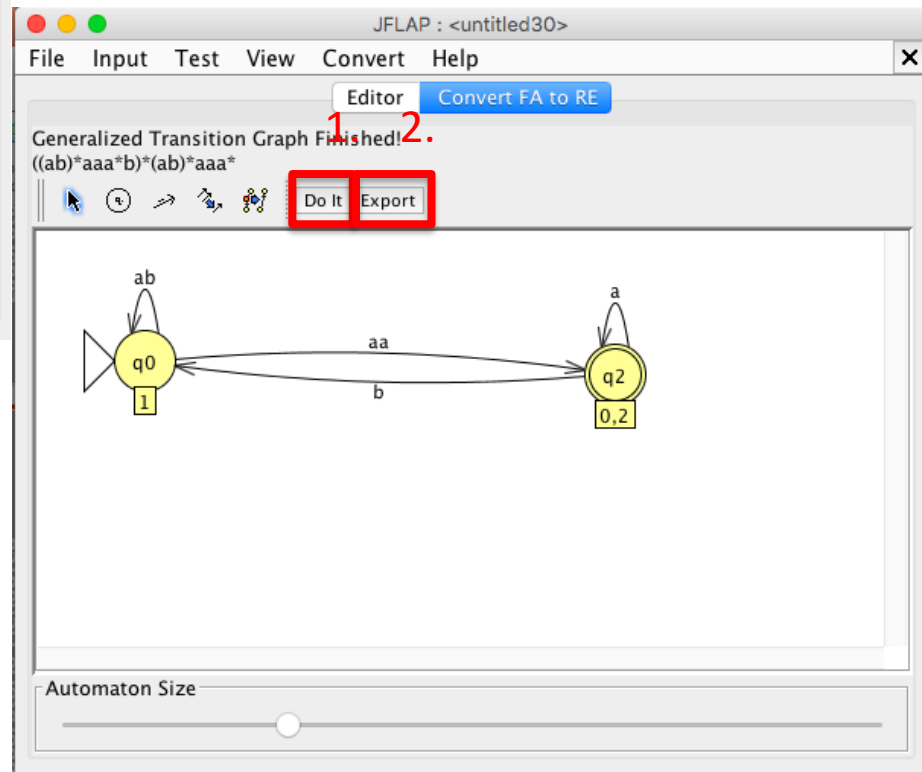
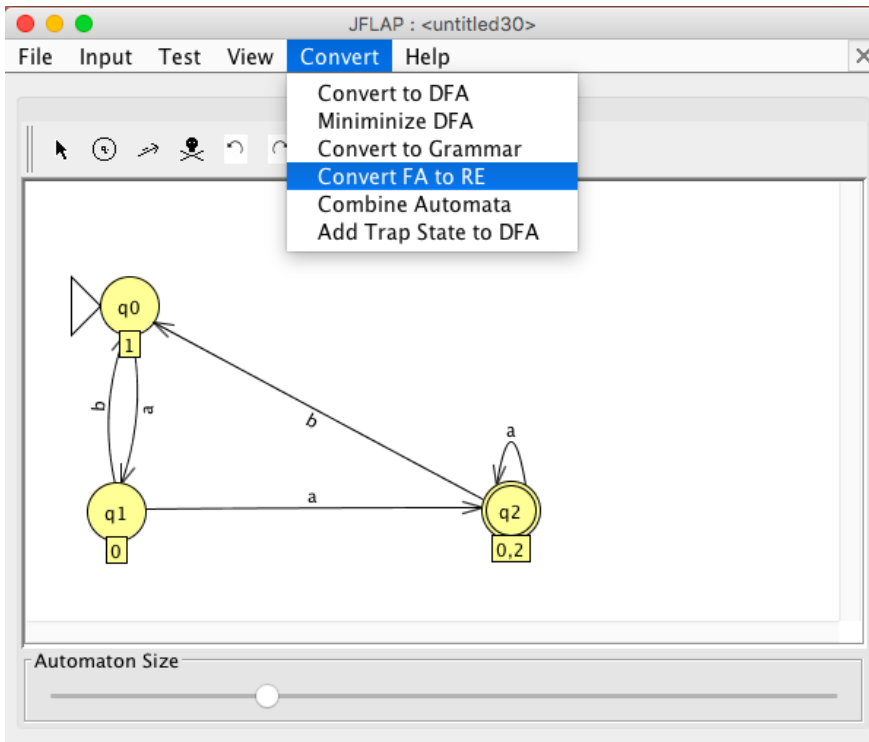
Production	Created
$S \rightarrow aB$	✓
$B \rightarrow aB$	✓
$B \rightarrow bS$	✓
$B \rightarrow a$	✓

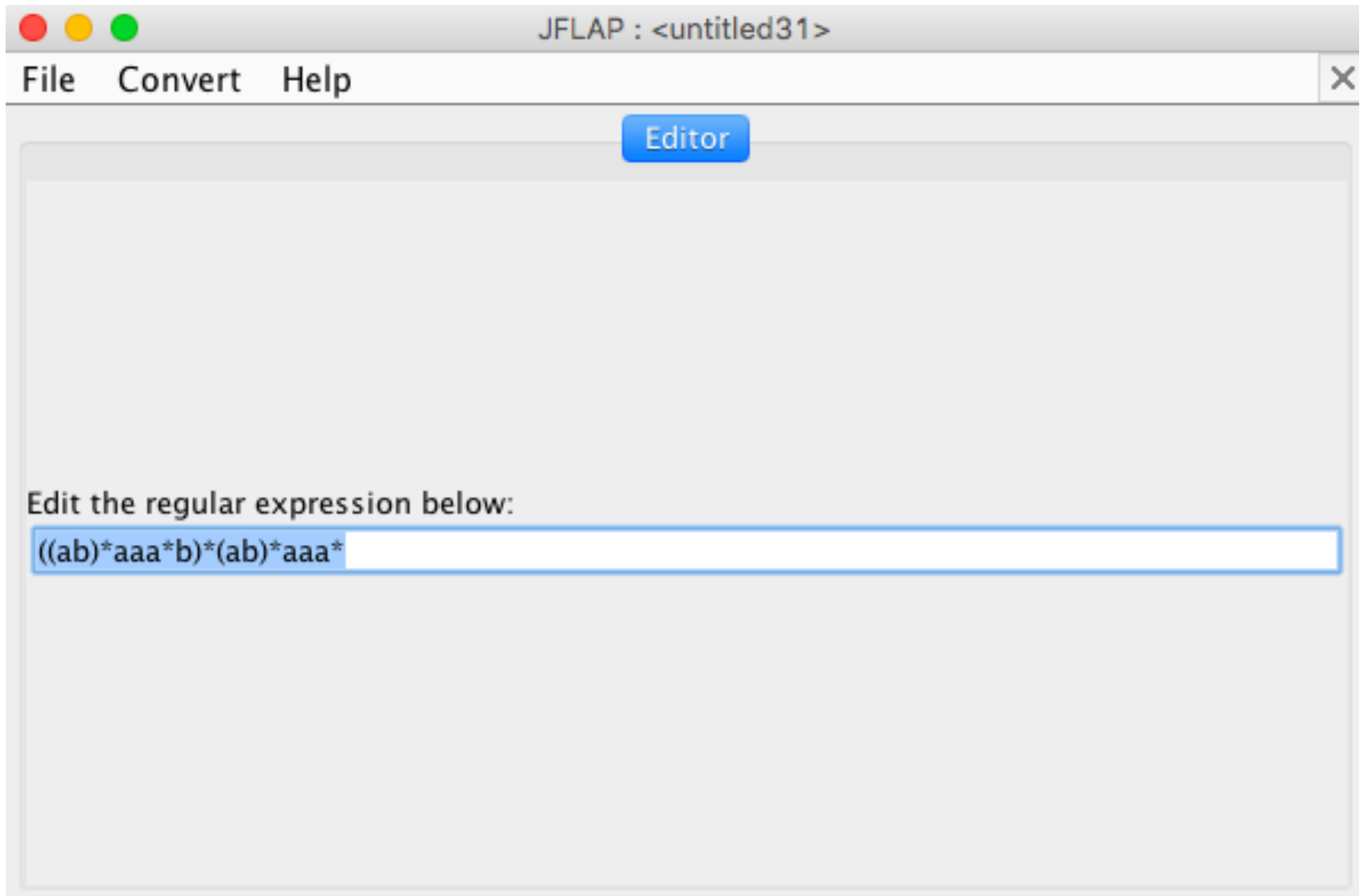
Show All Create Selected Done? **Export**

```
graph LR; start(( )) --> q1((q1)); q1 -- a --> q0(((q0))); q0 -- b --> q1; q0 -- a --> q0; q0 -- a --> q2(((q2))); style start fill:none,stroke:none
```

Automaton Size







Esempio A

RE: $a(a+b)^*+c$

JFLAP : <untitled19>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
a	Accept
c	Accept
ab	Accept
aa	Accept
aab	Accept
aba	Accept
abb	Accept
aaaba	Accept
ac	Reject
ca	Reject
b	Reject
ba	Reject
abc	Reject
bac	Reject
baa	Reject

Load Inputs Run Inputs Clear Enter Lambda

The DFA diagram shows three states: q0 (start), q1 (final), and q2. Transitions: q0 to q1 on 'c', q0 to q2 on 'a', q2 to q1 on 'b', and a self-loop on q2 for 'a'.

Esempio B

RE: $a(a+b)^*c$

JFLAP : <untitled19>

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
ac	Accept
aababac	Accept
abababc	Accept
abababc	Accept
abbbc	Accept
ca	Reject
babc	Reject
a	Reject

Load Inputs Run Inputs Clear Enter Lambda

The DFA diagram shows three states: q0 (start), q1 (final), and q2. Transitions: q0 to q2 on 'a', q2 to q1 on 'c', and a self-loop on q2 for 'b'.