Laboratorio Menabrea

dall'informatica di base alle applicazioni avanzate

JFLAP

prof. Giovanni Pani

Dott.ssa Vita Santa Barletta

JFLAP

- Software free per lo studio dell'informatica teorica
- In particolare
  - AUTOMI A STATI FINITI
  - LINGUAGGI FORMALI
- An Interactive Formal Languages and Automata Package

JFLAP

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

JFLAP

- Dal 1994 il software è stato convertito in Java (JFLAP)
- Funzionalità JFLAP
  - Conversione da automa a stati finiti non deterministico in automa a stati finiti deterministico, in grammatica formale o in espressione regolare
  - Creazione di automi a pila a partire da grammatiche context-free
  - Studio dei parser LR e SLR

JFLAP

- www.jflap.org

JFLAP

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

JFLAP

JFLAP Version 7.0
RELEASED August 28, 2009
Last update May 15, 2011

- Get JFLAP

JFLAP

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

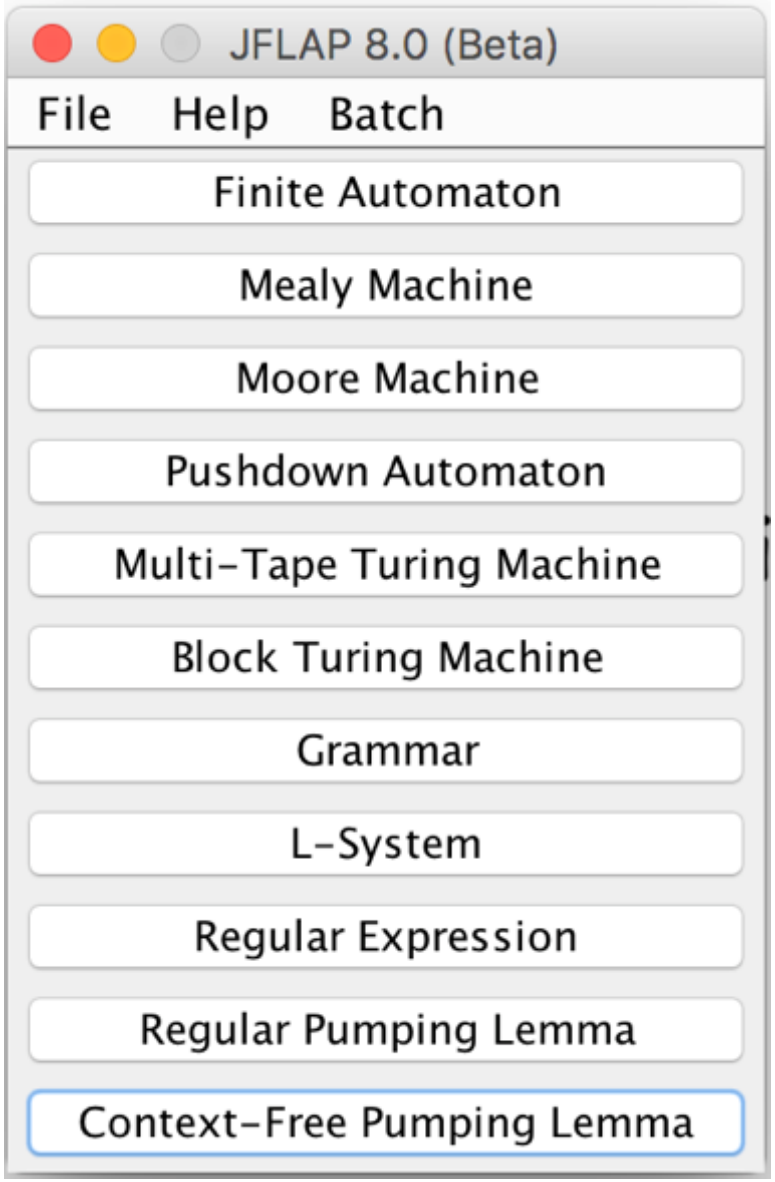
Get JFLAP

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.

JFLAP


JFLAP 8.0 (Beta)
File
Help
Batch
Finite Automaton
Mealy Machine
Moore Machine
Pushdown Automaton
Multi-Tape Turing Machine
Block Turing Machine
Grammar
L-System
Regular Expression
Regular Pumping Lemma
Context-Free Pumping Lemma

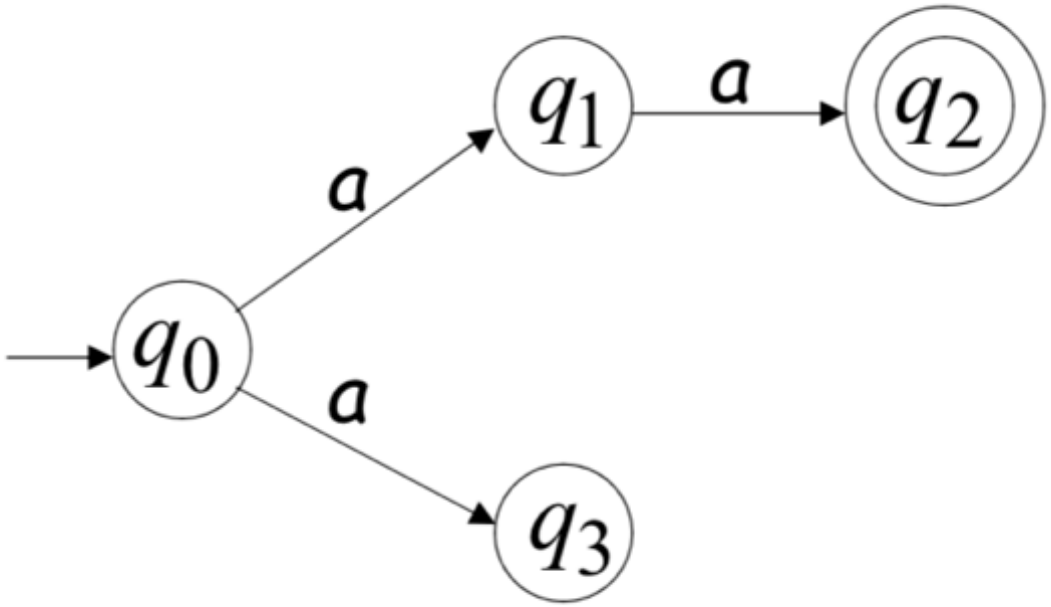
JFLAP: esempio

Conversione

NFA in DFA

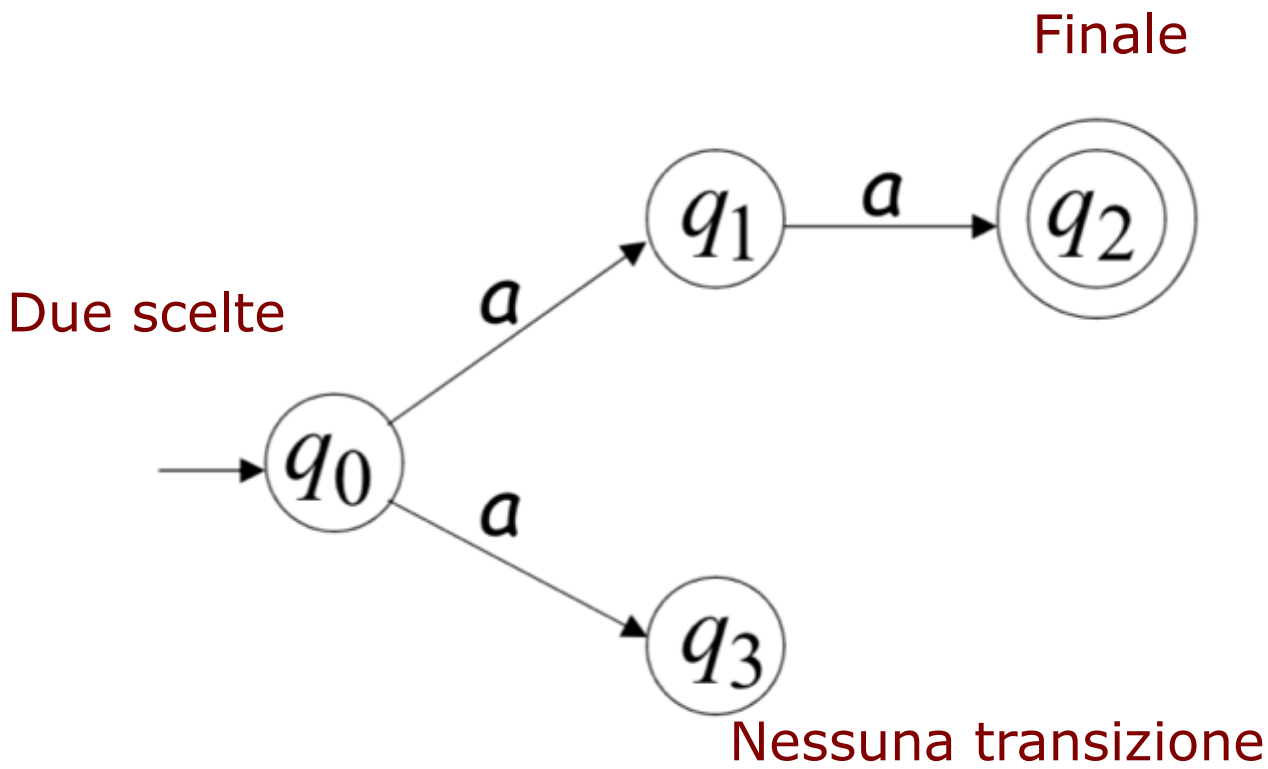
Automi non deterministici (NFA)

- Alfabeto={a}



Automi non deterministici (NFA)

- Alfabeto={a}

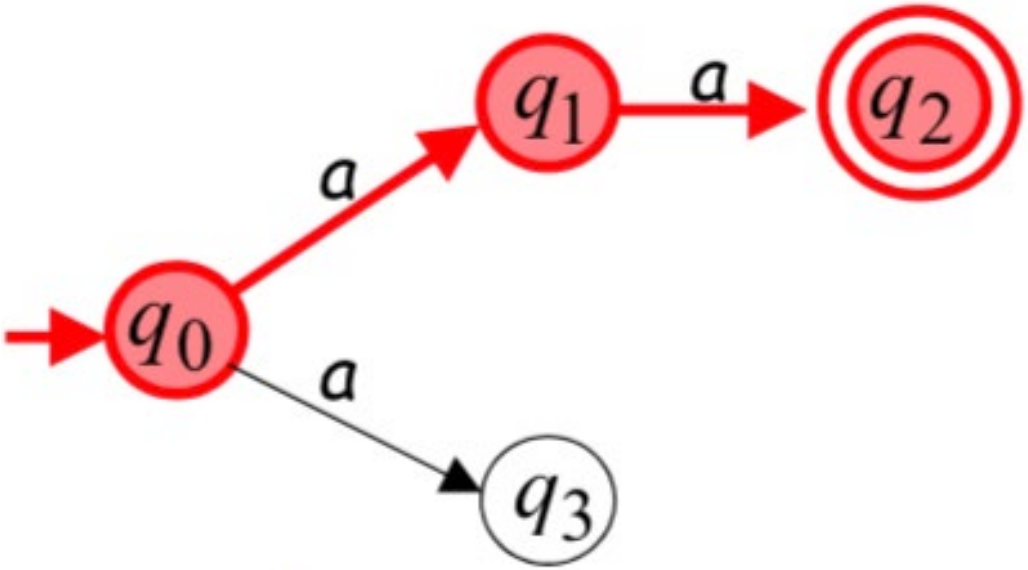


Automi non deterministici (NFA)

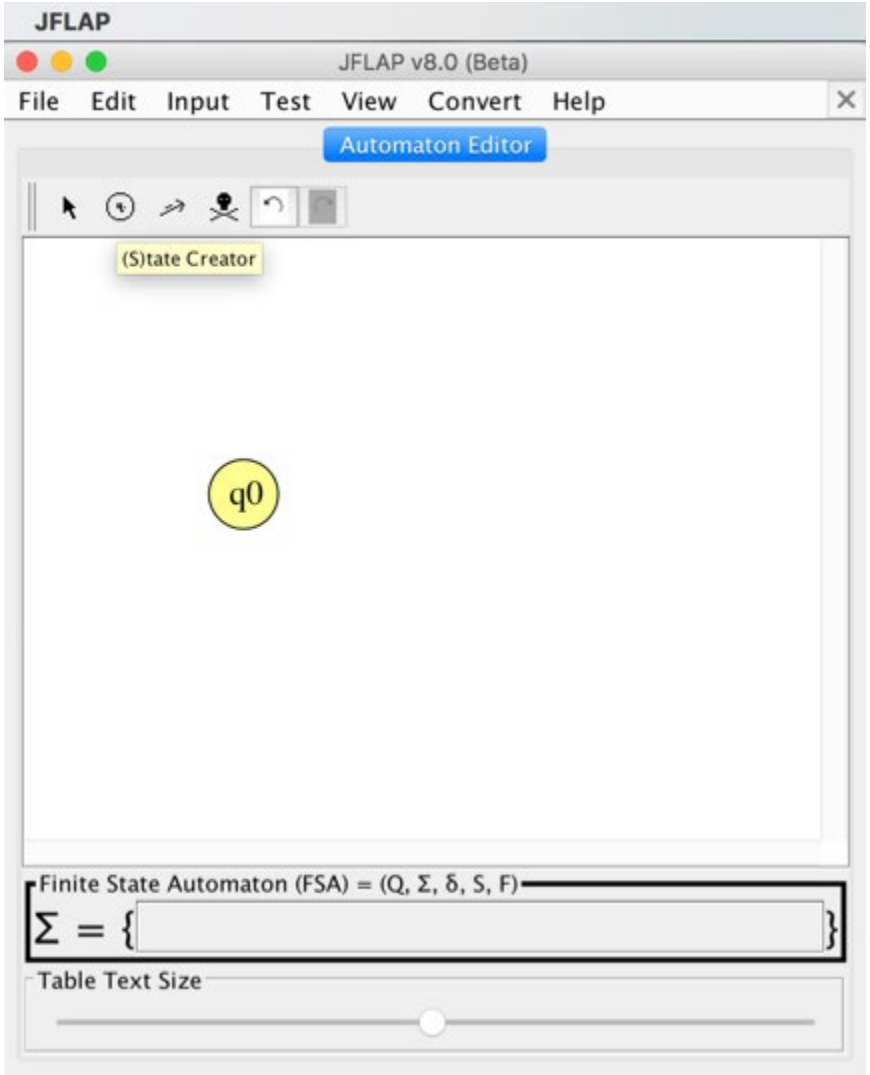
- NFA accetta una stringa se esiste una computazione che accetta la stringa
- Tutta la stringa di input è accettata e l'automa si trova in uno stato finale

Automi non deterministici (NFA)

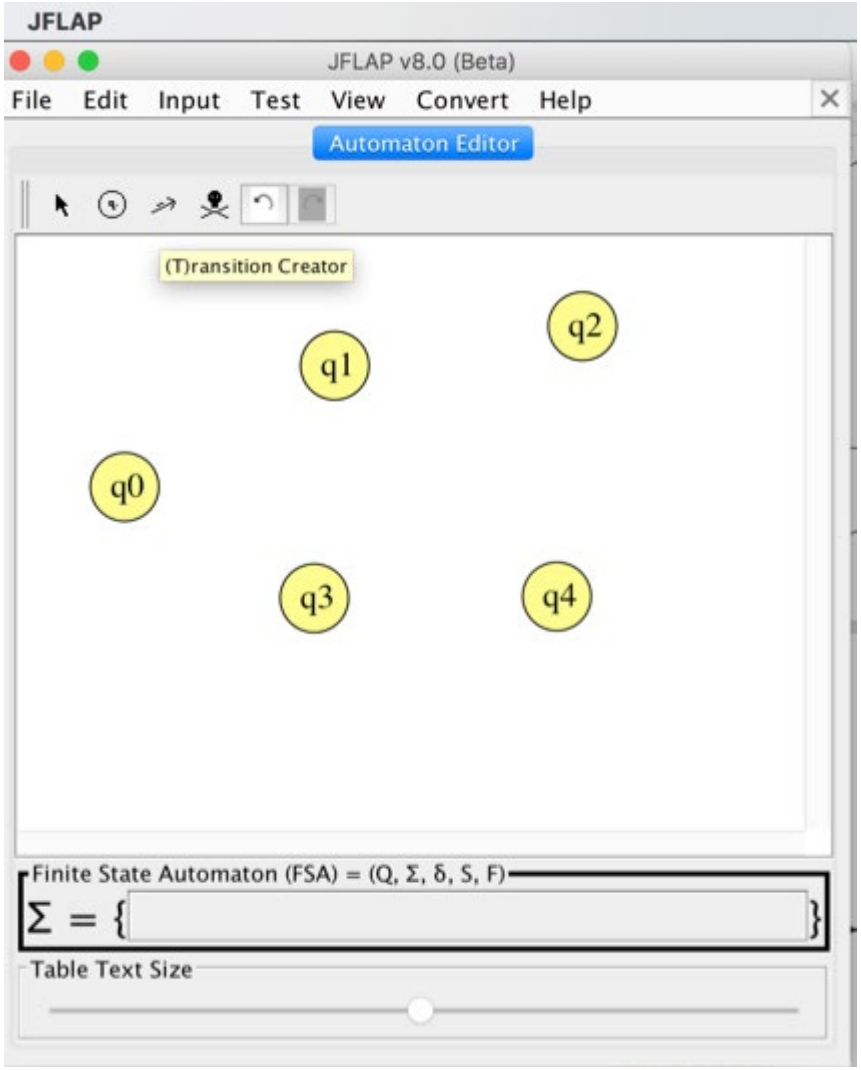
- La stringa *aa* è accettata dal NFA perché la computazione accetta *aa*



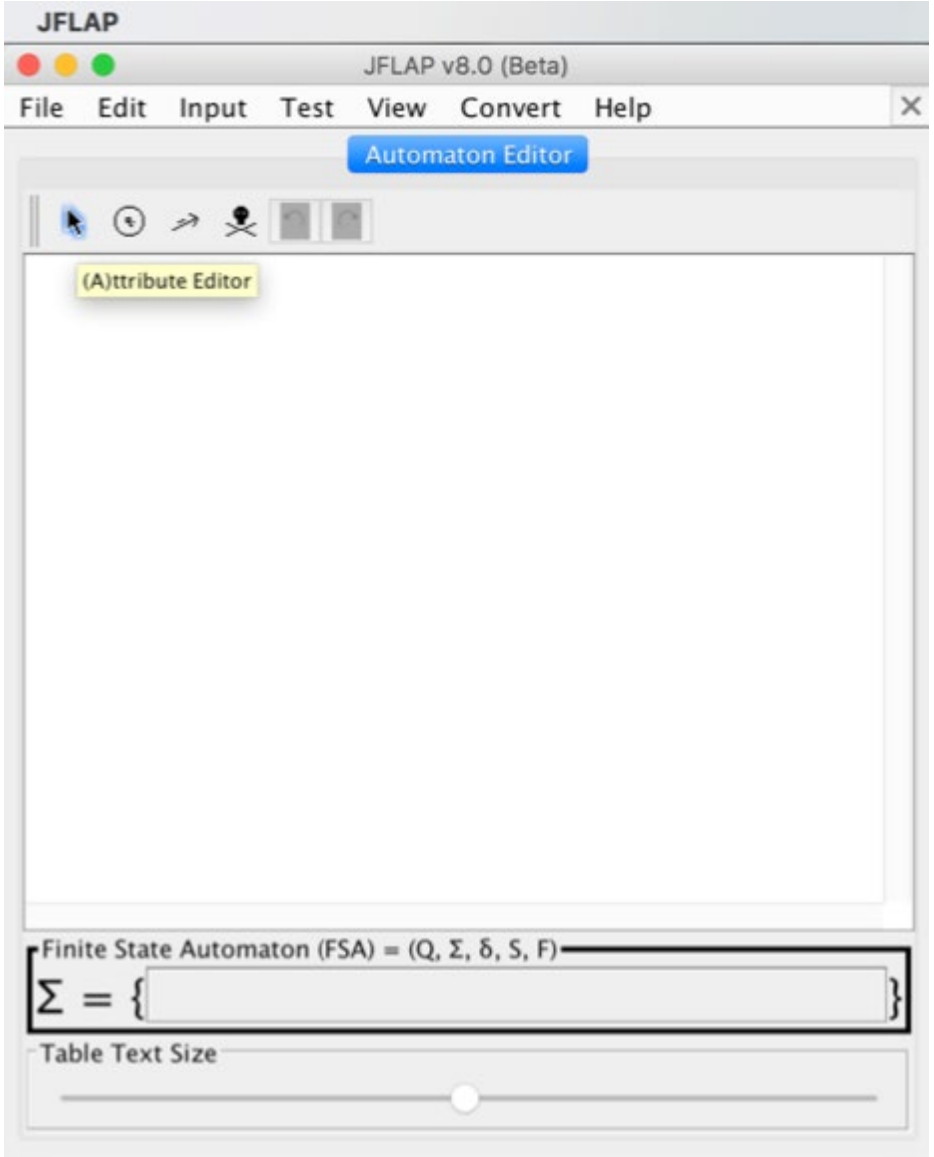
JFLAP: NFA


JFLAP
JFLAP v8.0 (Beta)
File
Edit
Input
Test
View
Convert
Help
Automaton Editor
(S)tate Creator
q0
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = {
}
Table Text Size

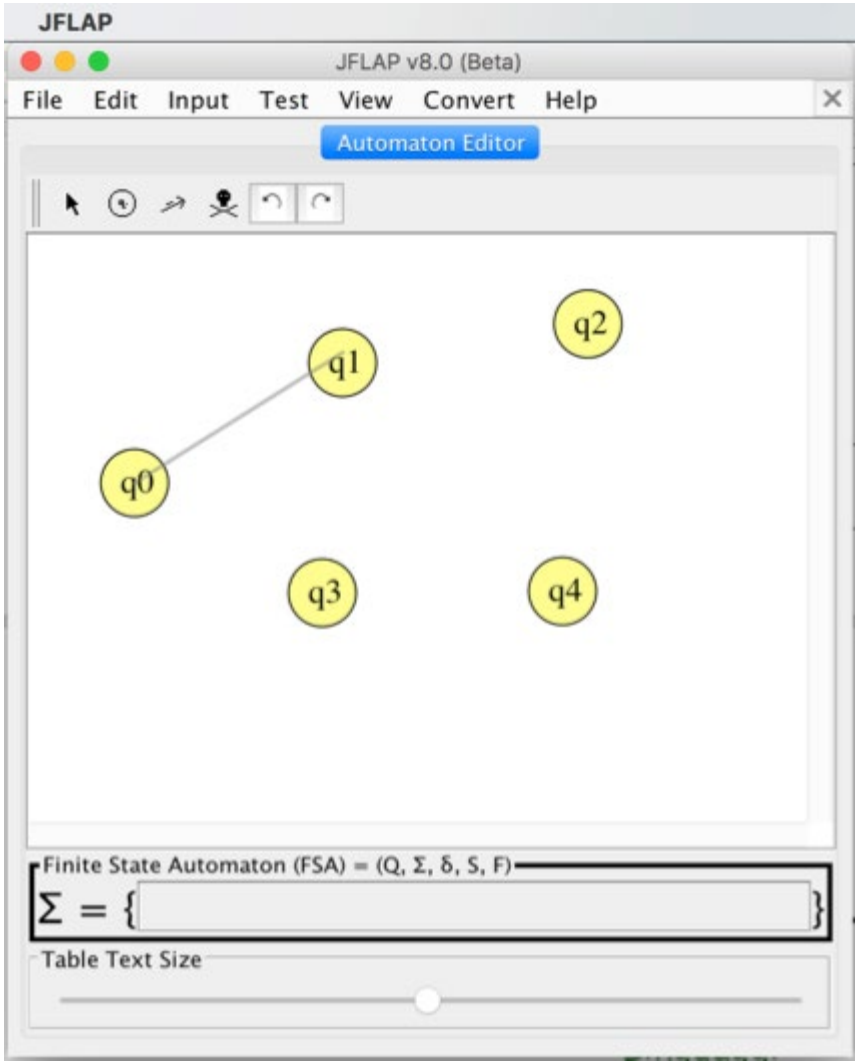
JFLAP: NFA

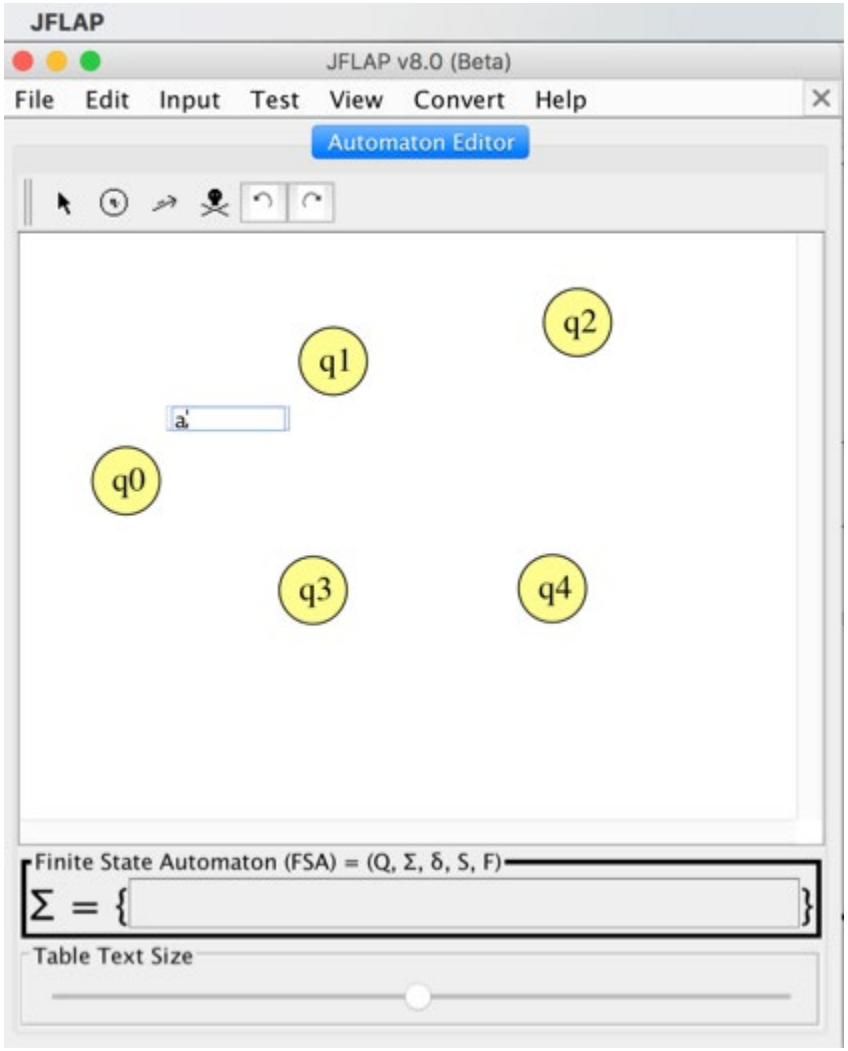

JFLAP
JFLAP v8.0 (Beta)
File
Edit
Input
Test
View
Convert
Help
Automaton Editor
(T)ransition Creator
q0
q1
q2
q3
q4
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = {
}
Table Text Size

JFLAP: NFA

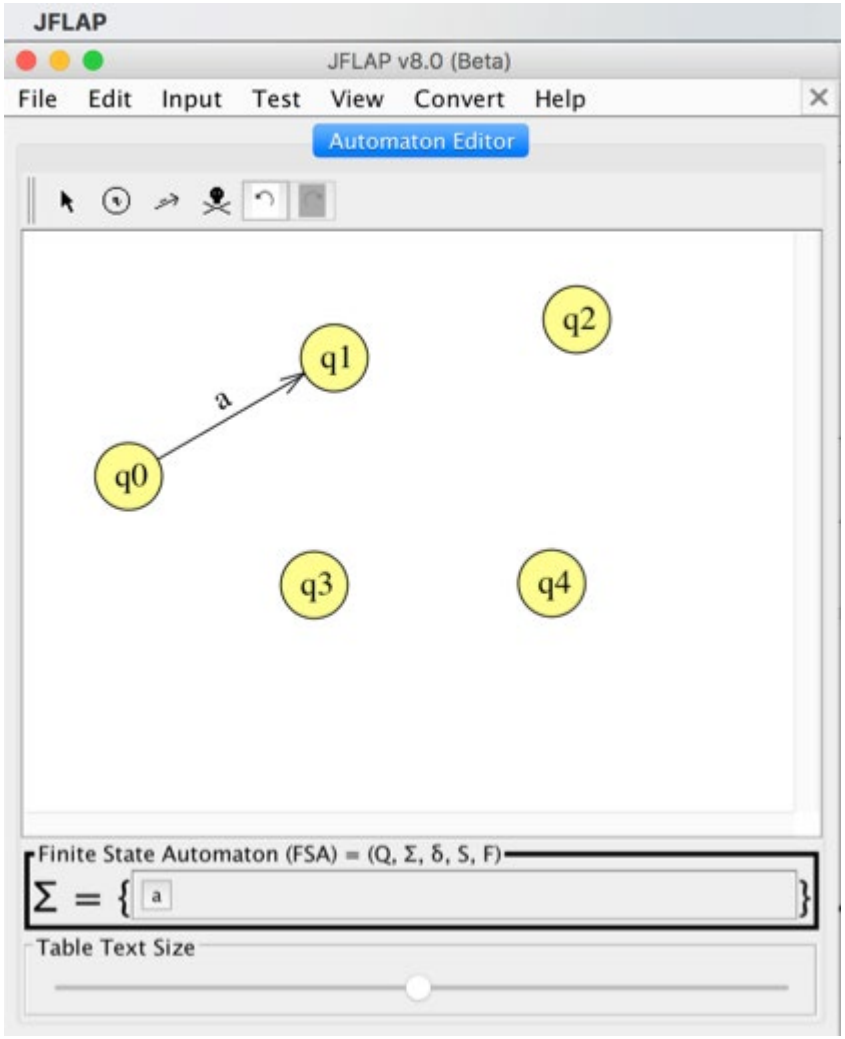

JFLAP
JFLAP v8.0 (Beta)
File
Edit
Input
Test
View
Convert
Help
Automaton Editor
(A)ttribute Editor
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = {
}
Table Text Size

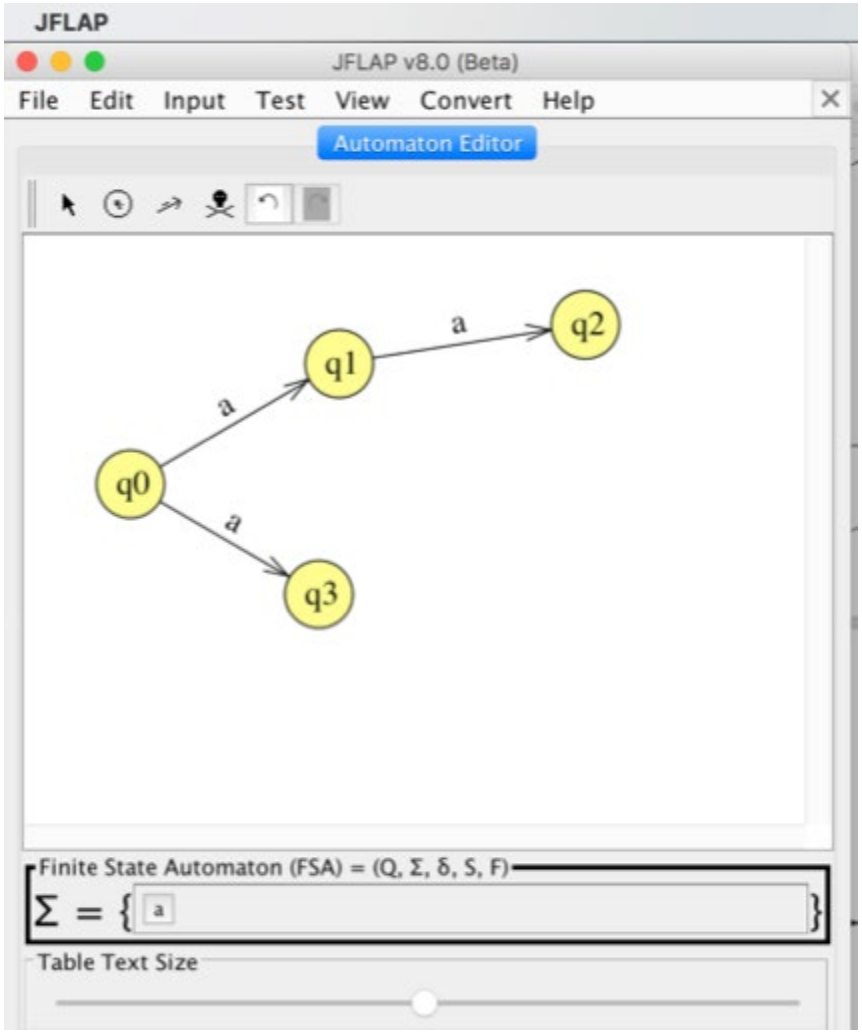
JFLAP: NFA


JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
q4
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = {
Table Text Size

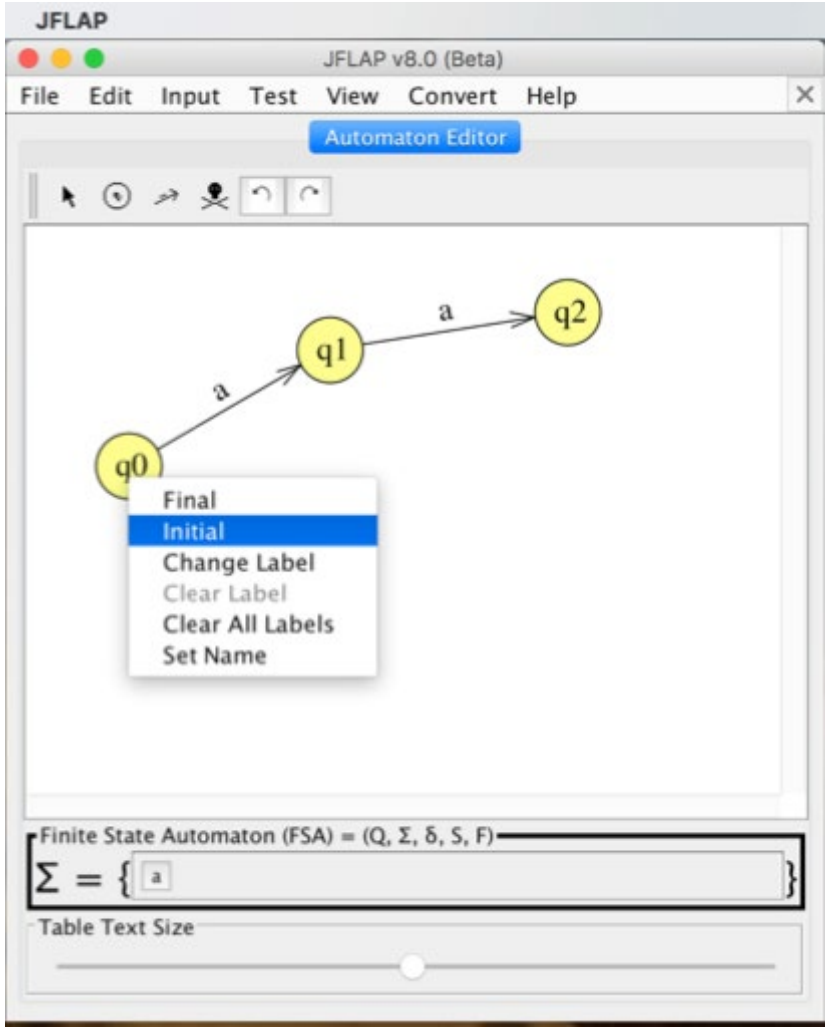

JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
q4
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = {
Table Text Size

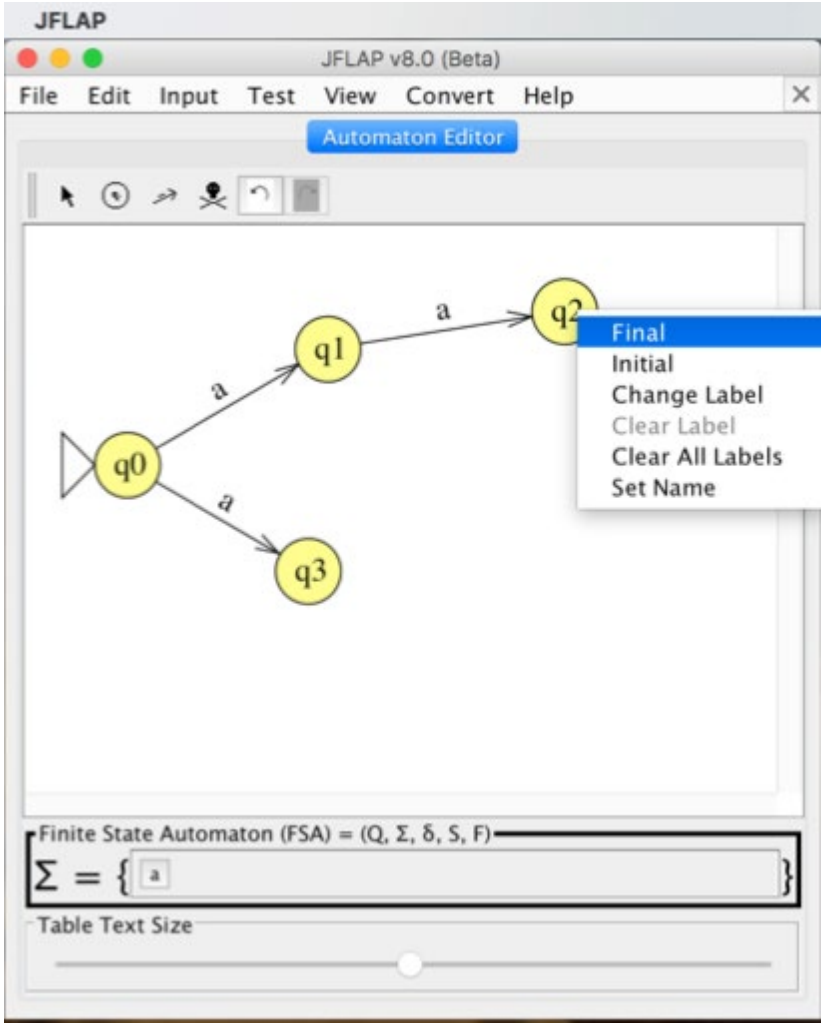
JFLAP: NFA


JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
q4
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

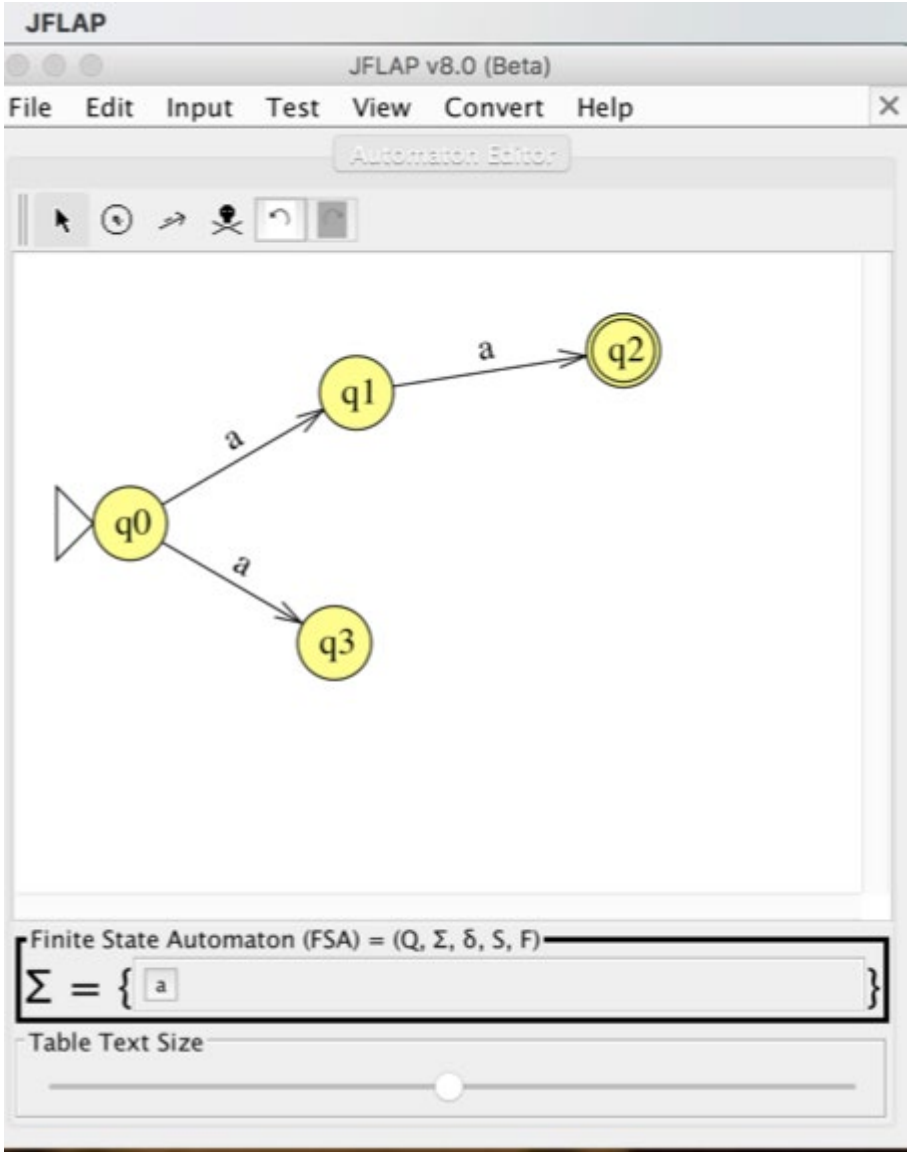

JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
a
a
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

JFLAP: NFA

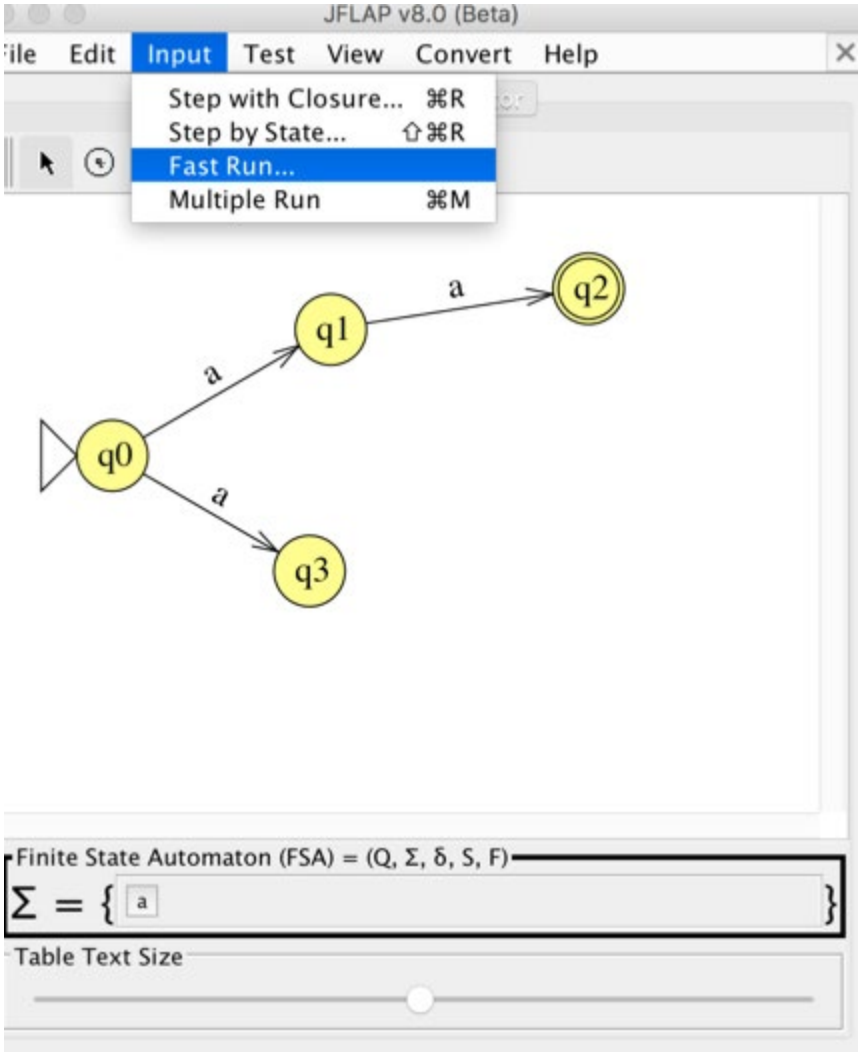

JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
a
a
Final
Initial
Change Label
Clear Label
Clear All Labels
Set Name
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size


JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
a
a
a
Final
Initial
Change Label
Clear Label
Clear All Labels
Set Name
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

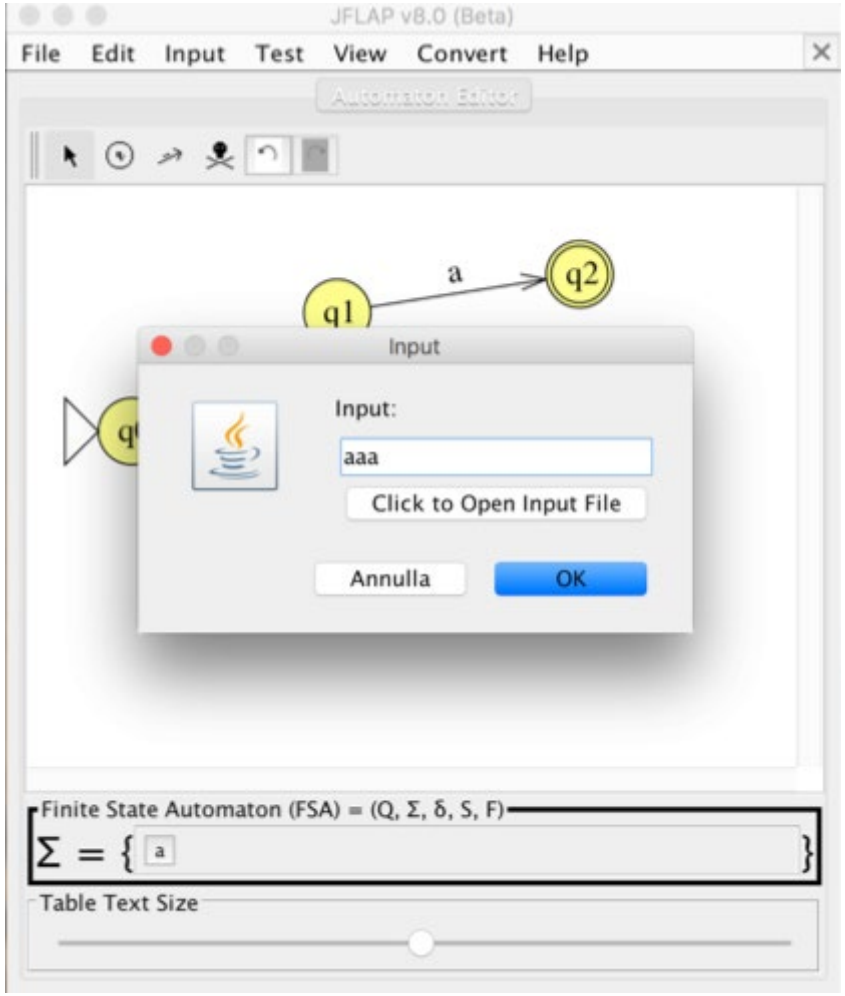
JFLAP: NFA

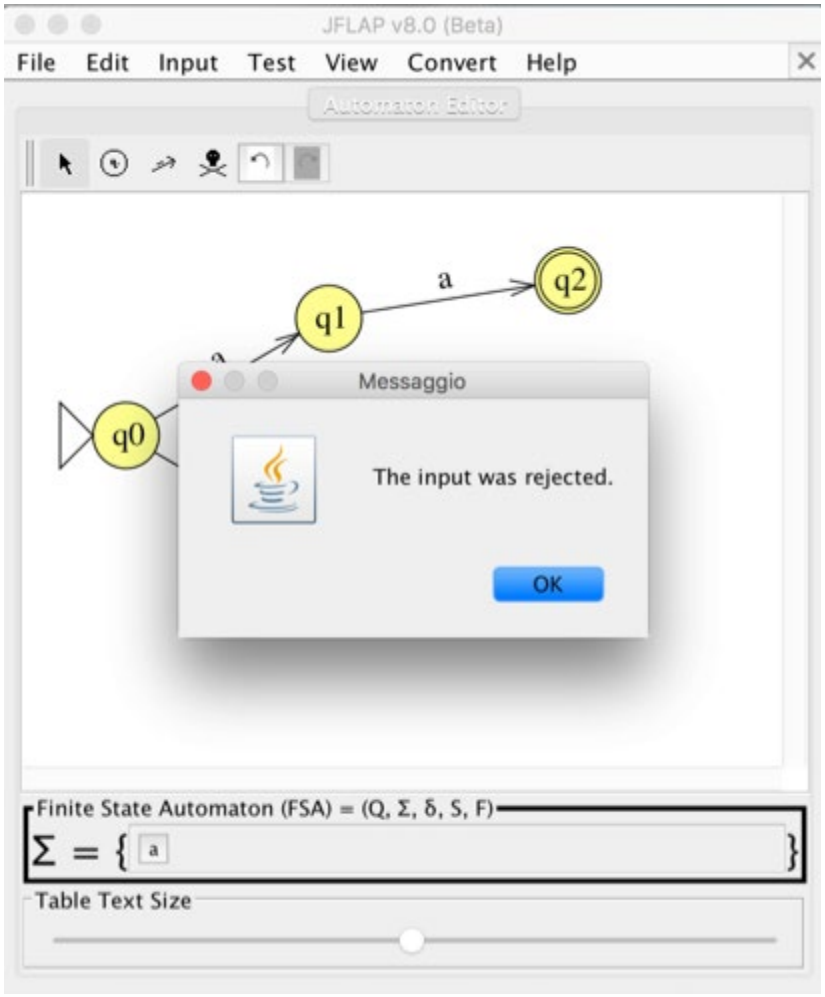

JFLAP
JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
q2
q3
a
a
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

JFLAP: NFA

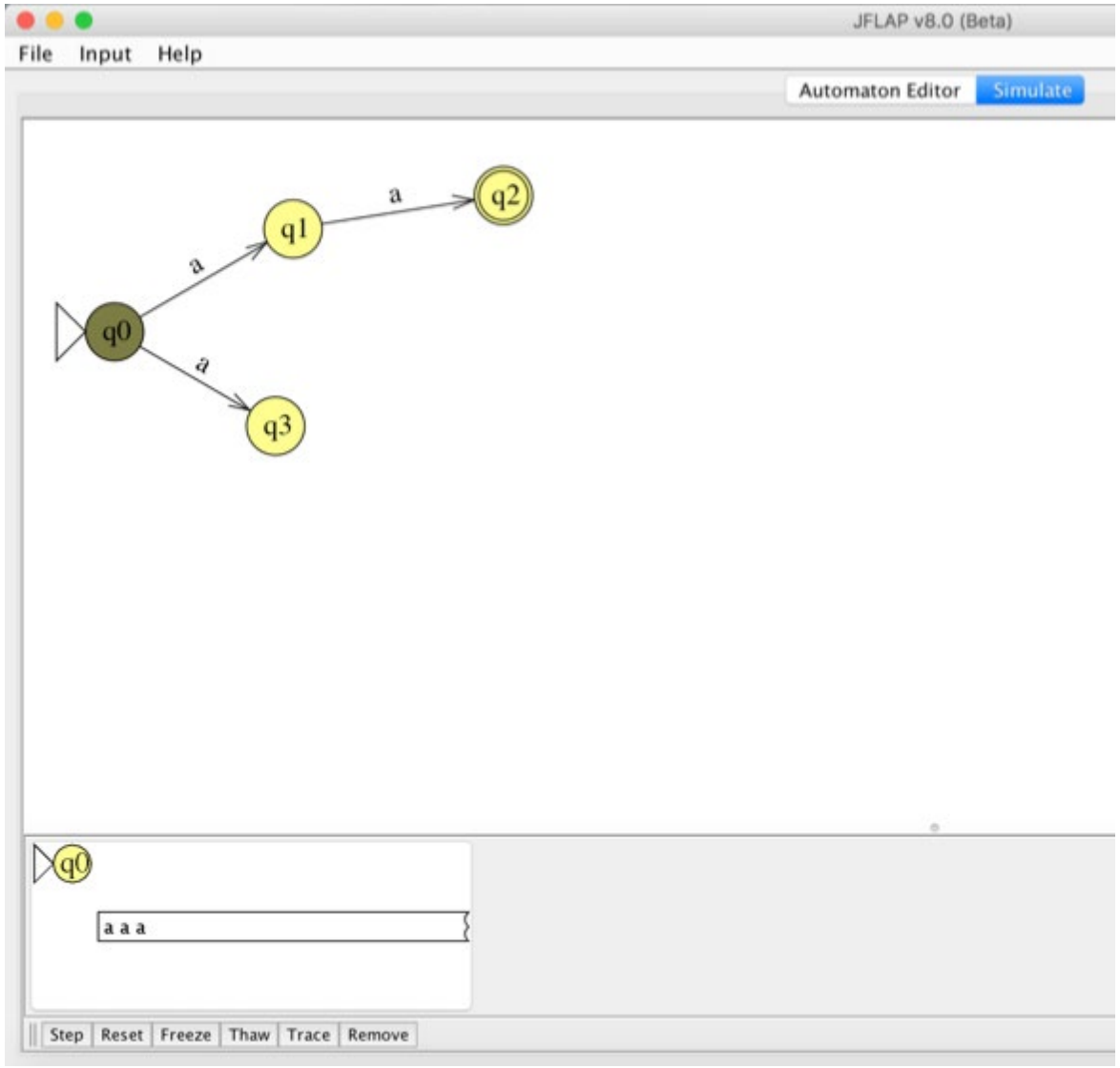

JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Step with Closure... ⌘R
Step by State... ⇧⌘R
Fast Run...
Multiple Run ⌘M
q0
q1
q2
q3
a
a
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

JFLAP: NFA

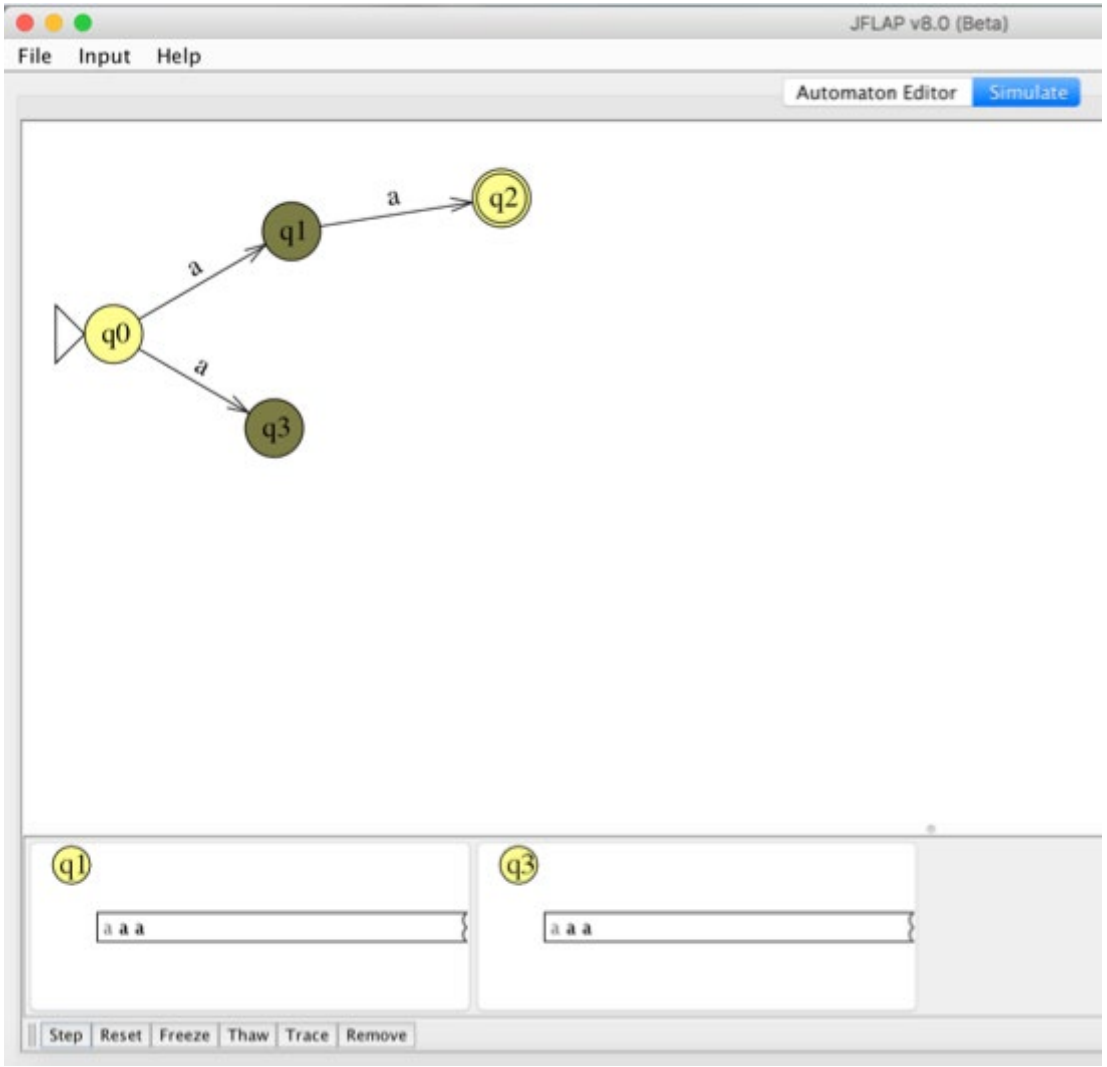

JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q1
a
q2
Input
Input:
aaa
Click to Open Input File
Annulla
OK
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size


JFLAP v8.0 (Beta)
File Edit Input Test View Convert Help
Automaton Editor
q0
q1
a
q2
Messaggio
The input was rejected.
OK
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

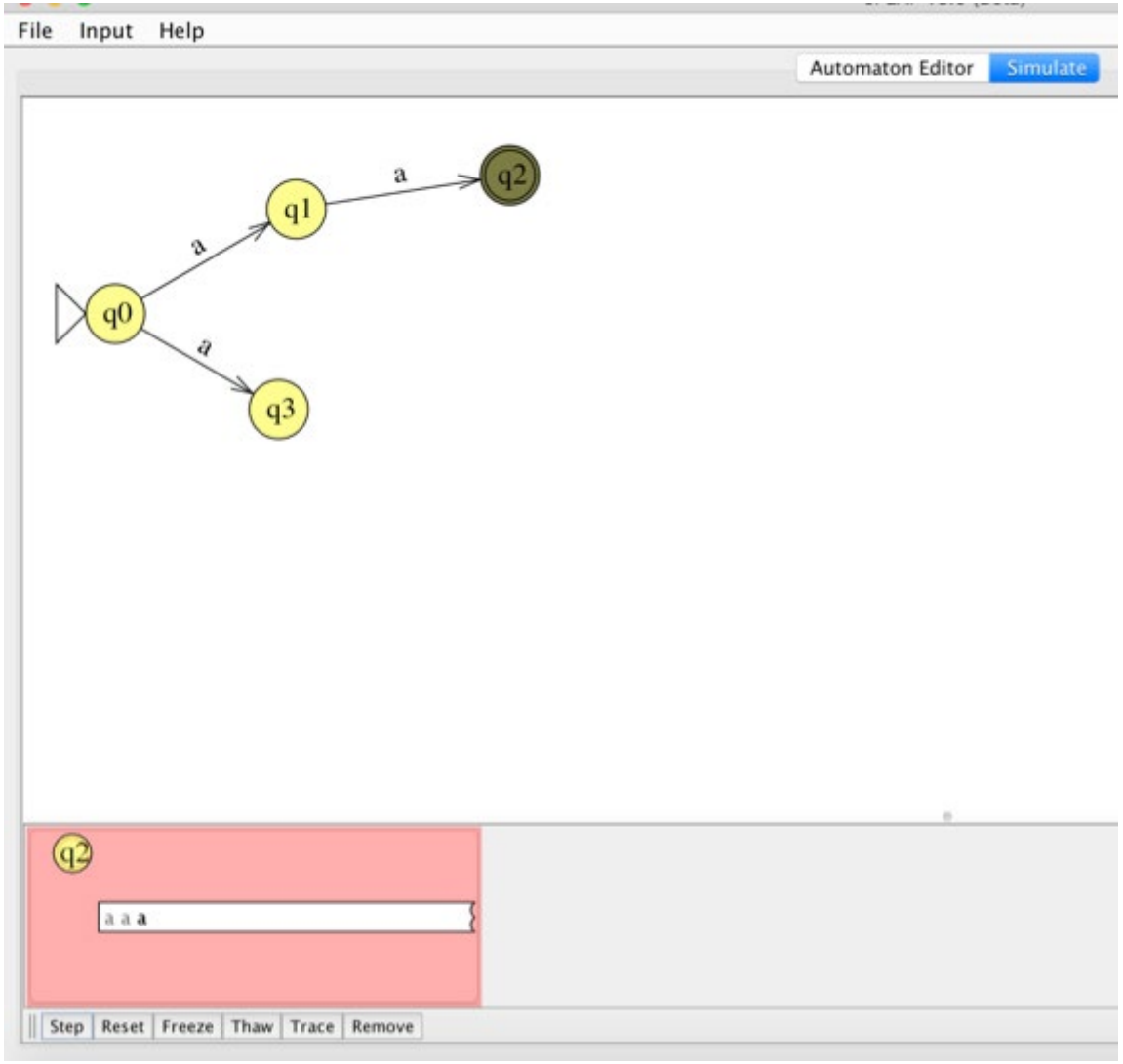
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Simulate
q0
q1
q2
q3
a
a
a
q0
a a a
Step
Reset
Freeze
Thaw
Trace
Remove

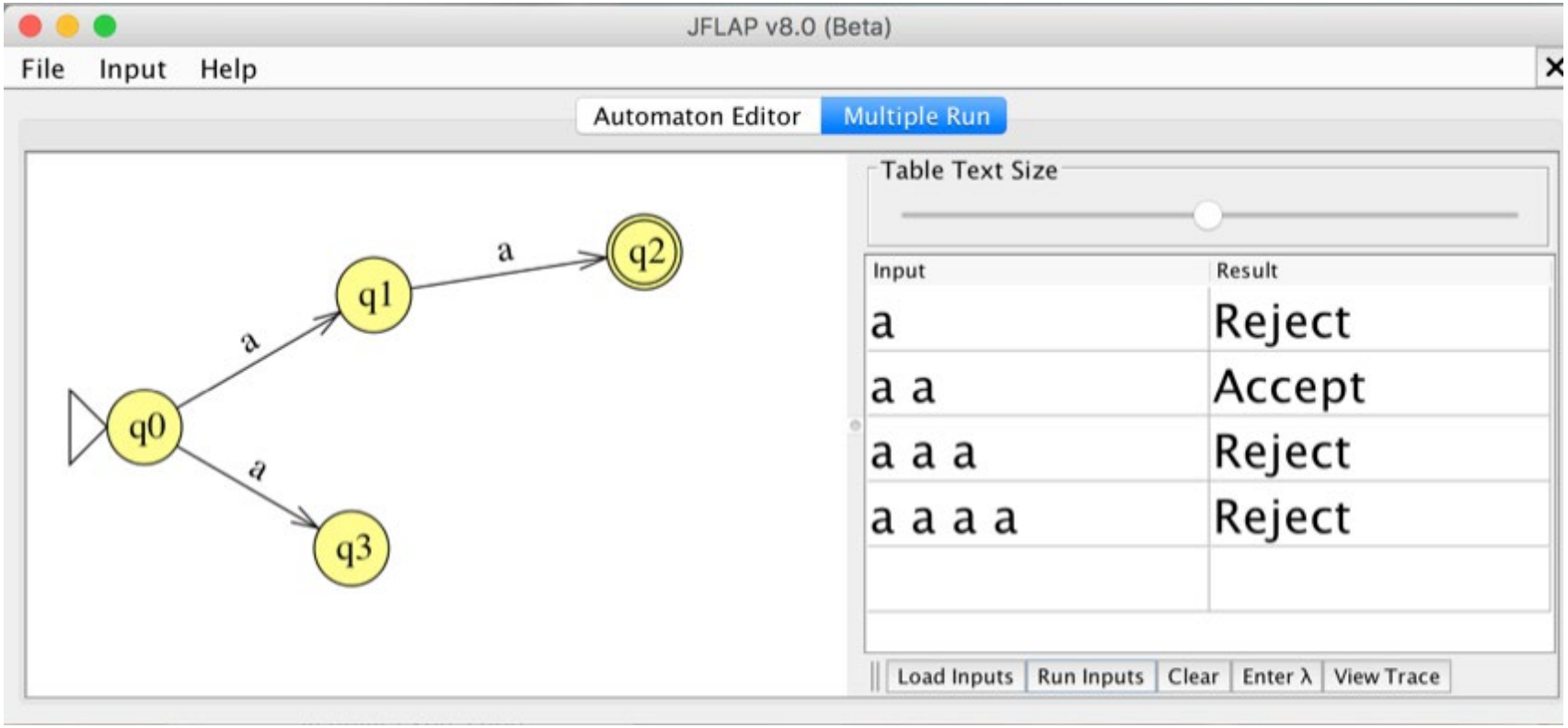
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Simulate
q0
q1
q2
q3
a
a
a
q1
a a a
q3
a a a
Step
Reset
Freeze
Thaw
Trace
Remove

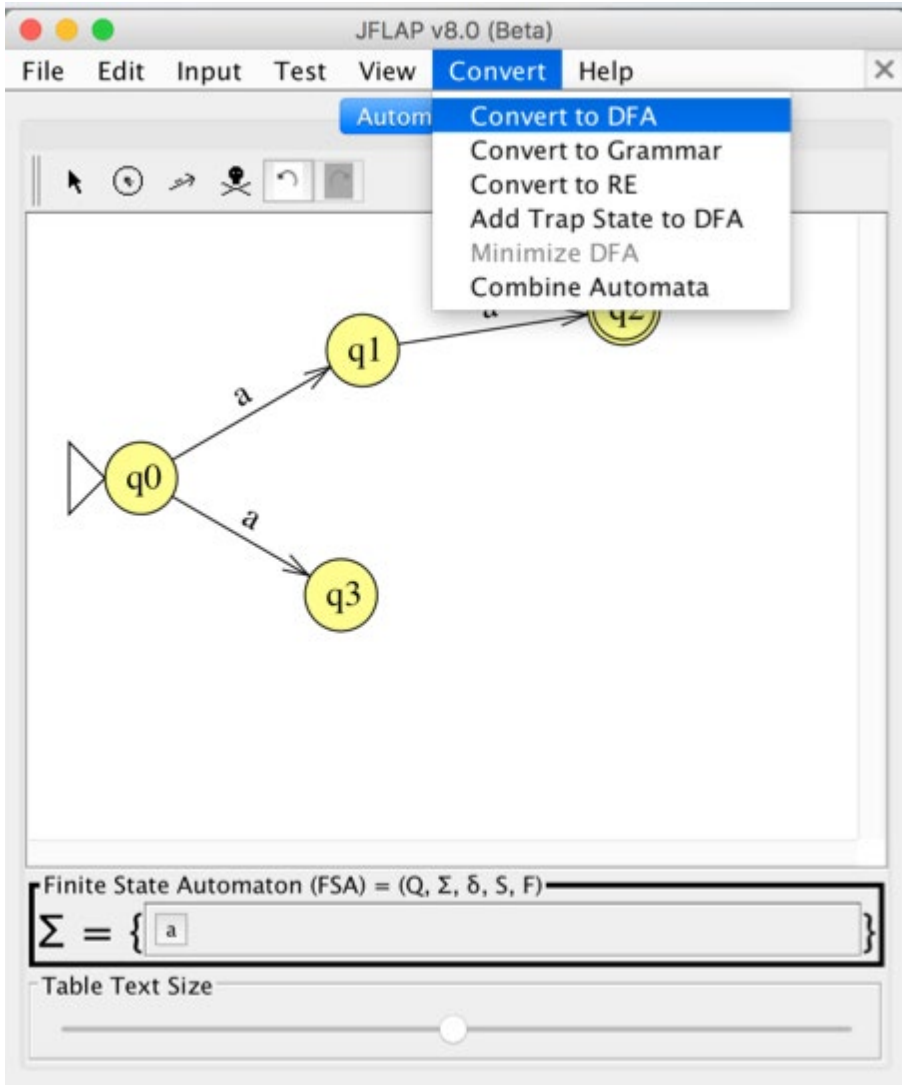
JFLAP: NFA


File
Input
Help
Automaton Editor
Simulate
q0
q1
q2
q3
a
a
a
q2
a a a
Step
Reset
Freeze
Thaw
Trace
Remove

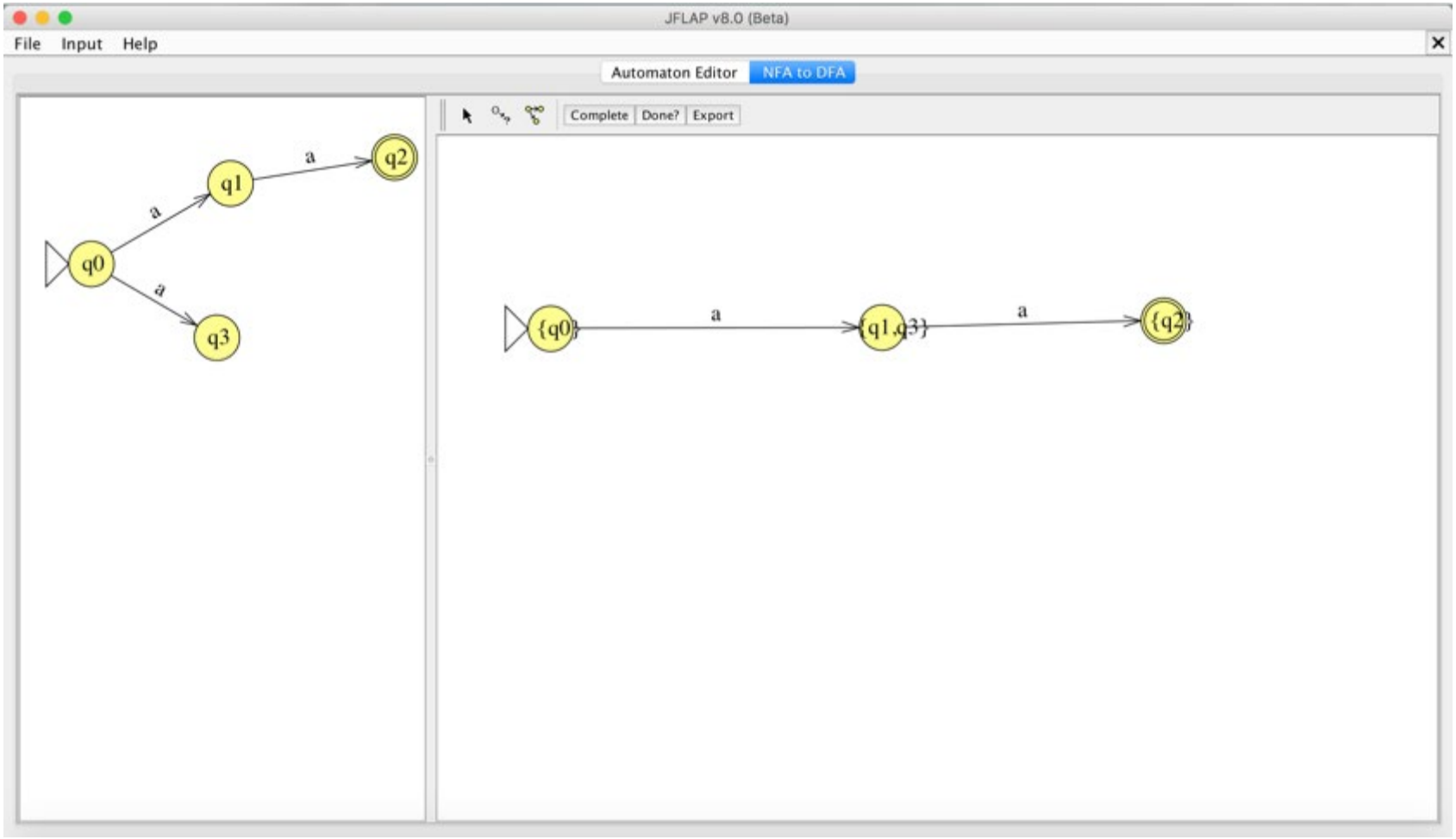
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Multiple Run
Table Text Size
q0
q1
q2
q3
a
a
a
Input
Result
a
Reject
a a
Accept
a a a
Reject
a a a a
Reject
Load Inputs
Run Inputs
Clear
Enter λ
View Trace

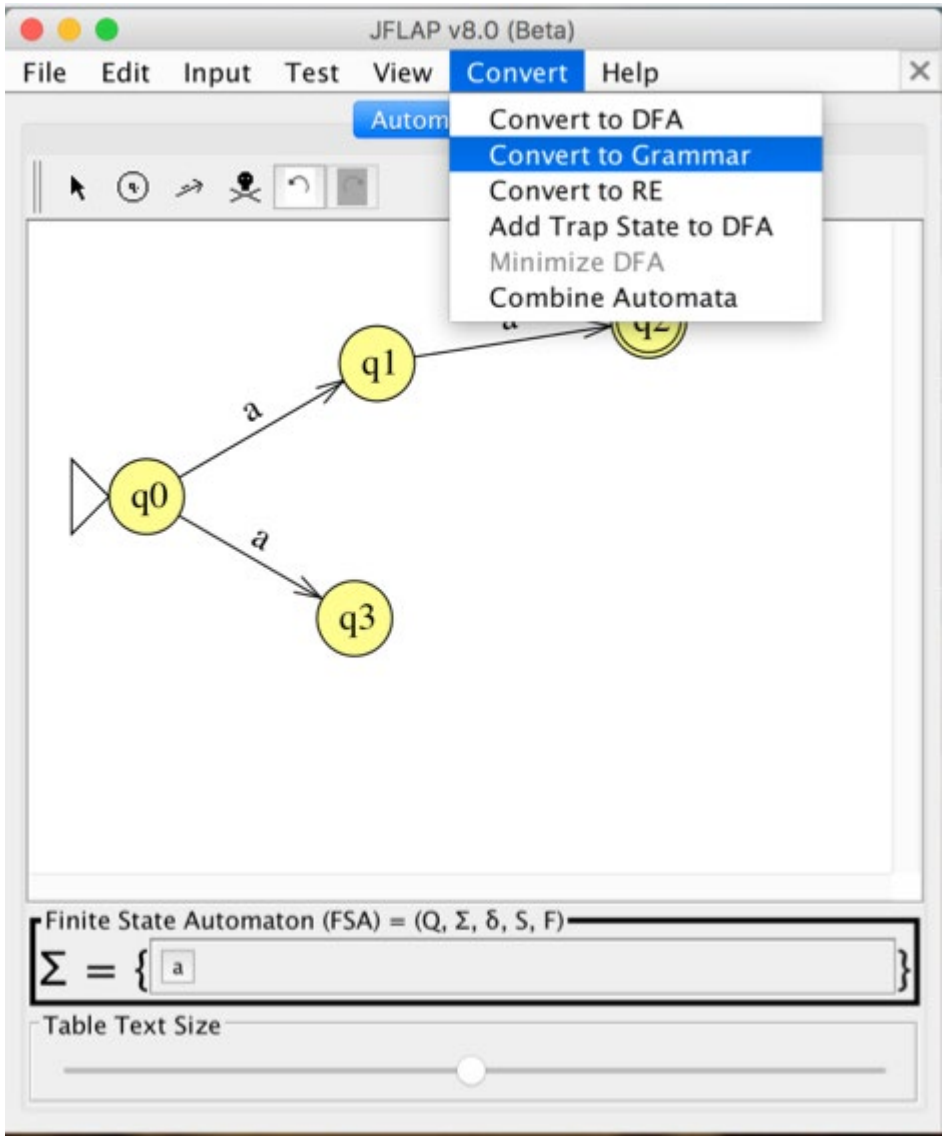
Conversione NFA in DFA



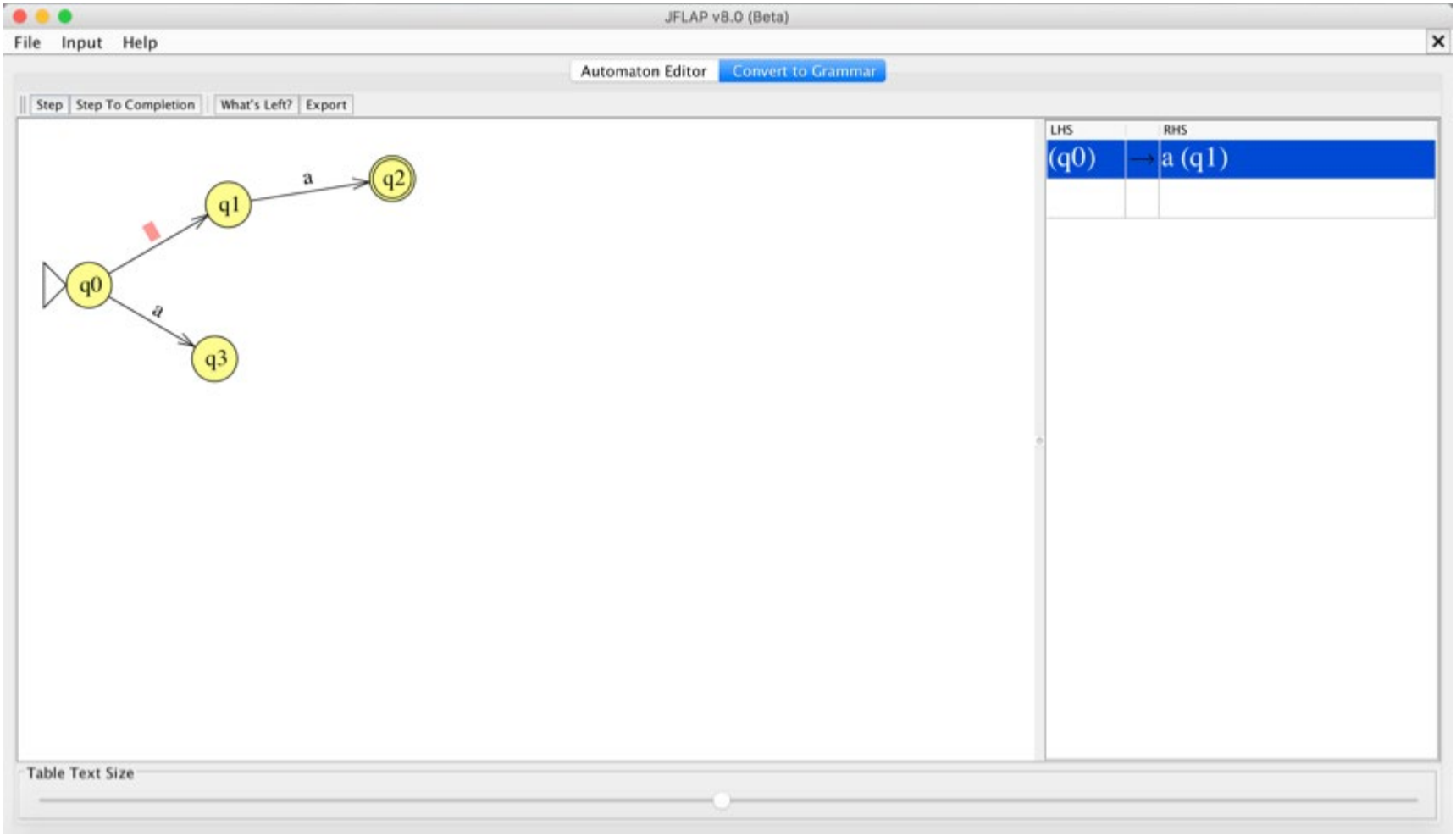
Conversione NFA in DFA



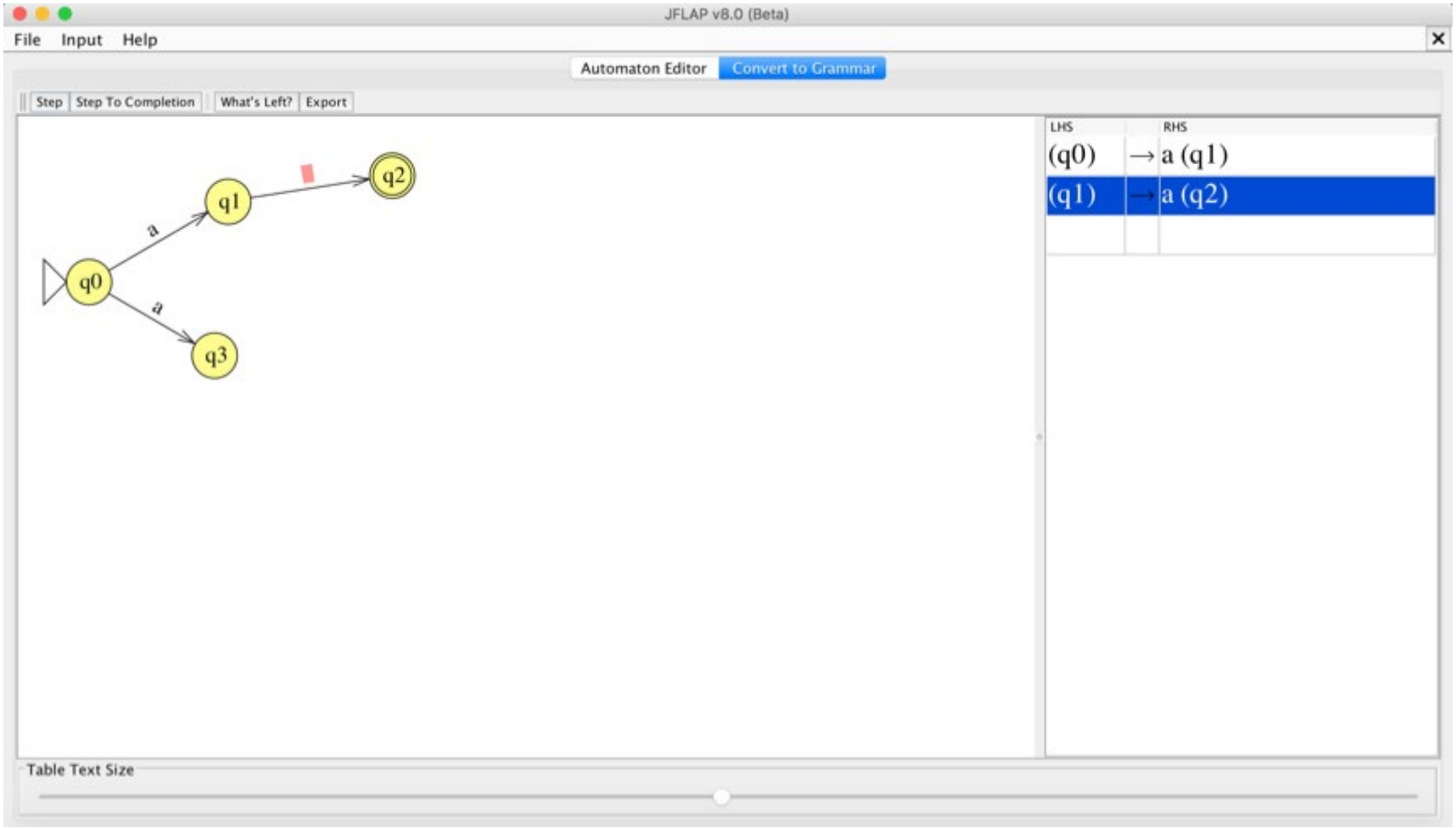
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Edit
Input
Test
View
Convert
Help
Convert to DFA
Convert to Grammar
Convert to RE
Add Trap State to DFA
Minimize DFA
Combine Automata
q0
q1
q2
q3
a
a
Finite State Automaton (FSA) = (Q, Σ, δ, S, F)
Σ = { a }
Table Text Size

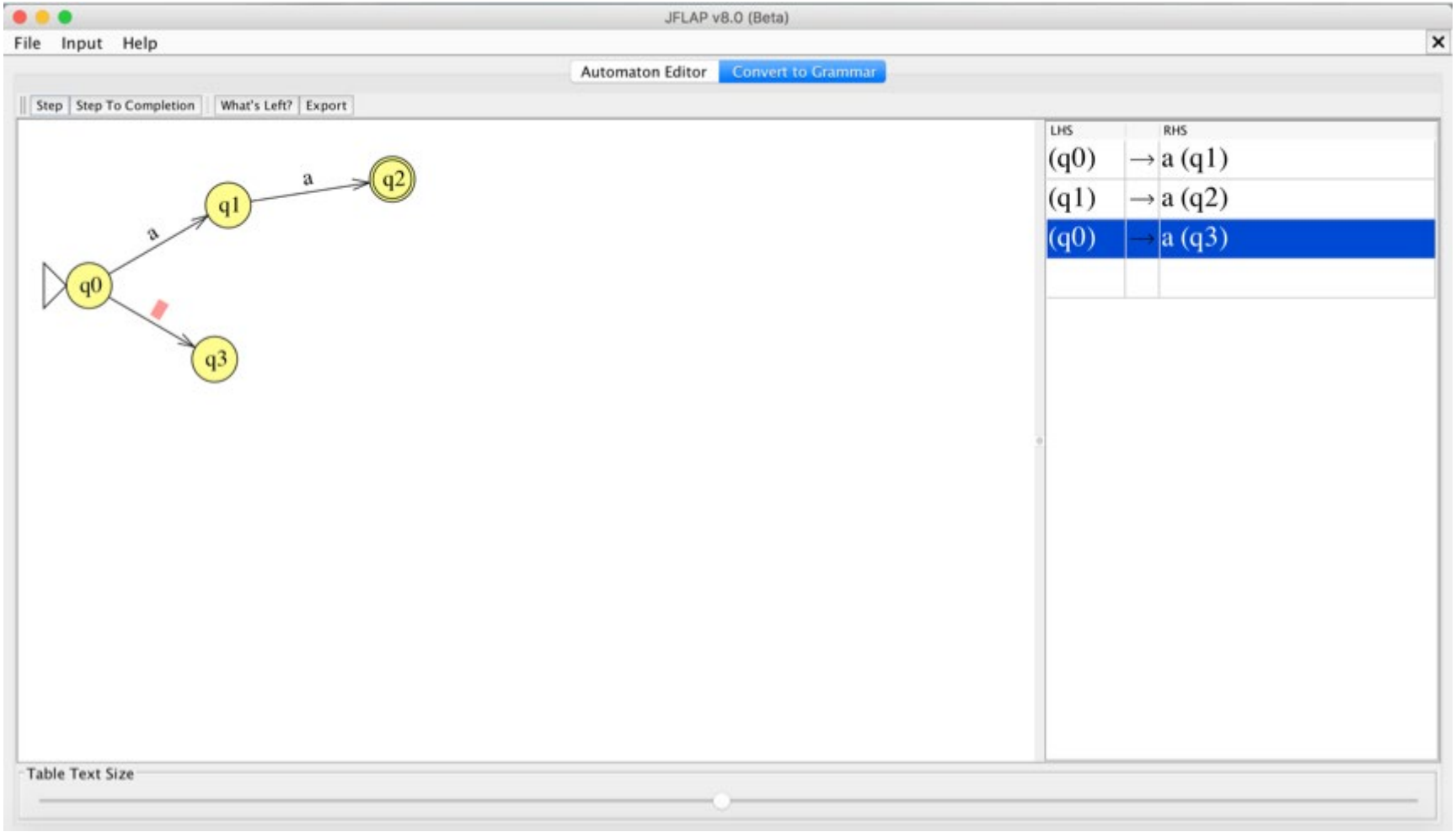
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Convert to Grammar
Step
Step To Completion
What's Left?
Export
q0
q1
q2
q3
a
a
LHS
RHS
(q0)
→
a (q1)
Table Text Size

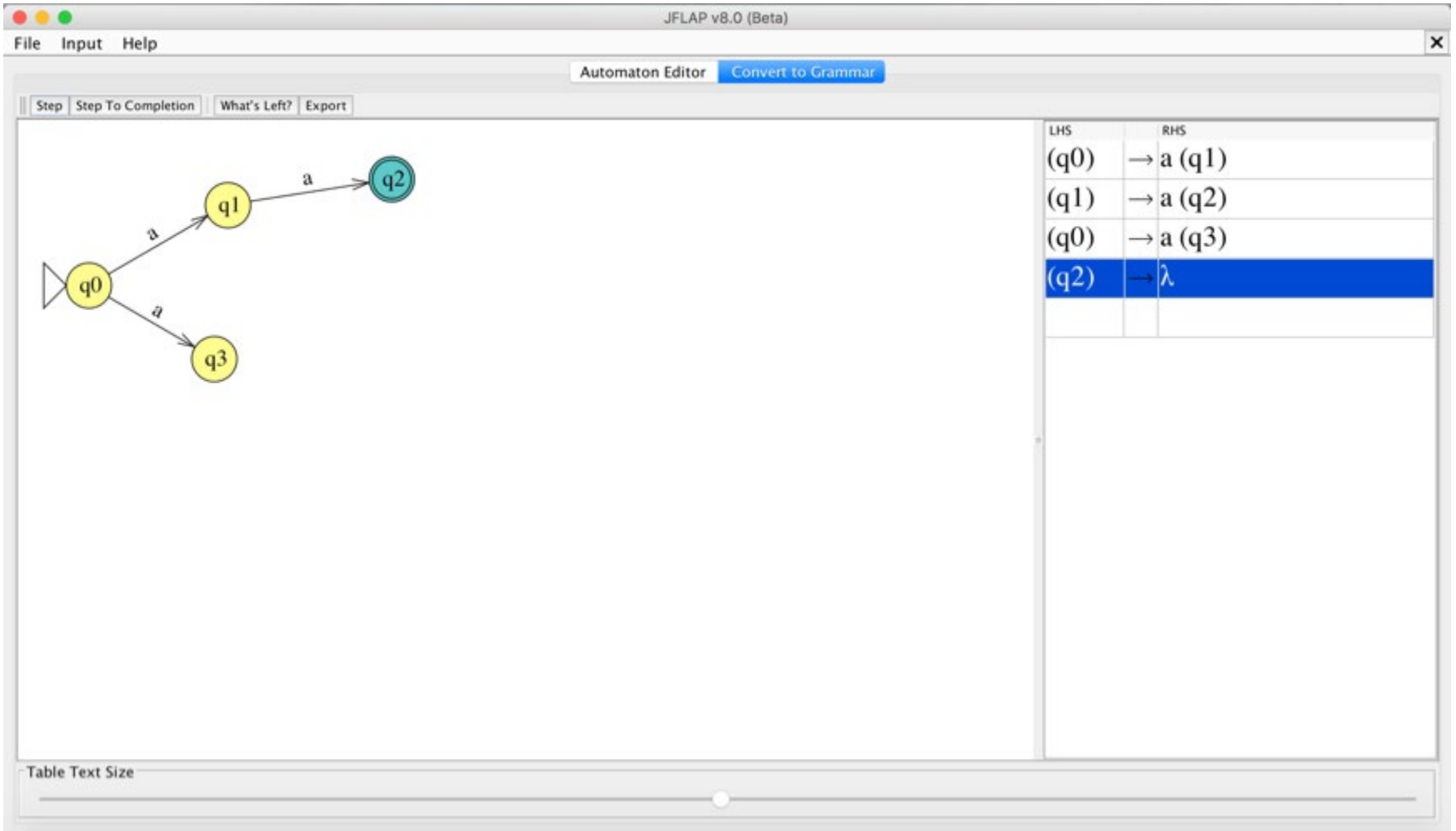
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Convert to Grammar
Step
Step To Completion
What's Left?
Export
q0
q1
q2
q3
a
a
LHS
RHS
(q0) → a (q1)
(q1) → a (q2)
Table Text Size

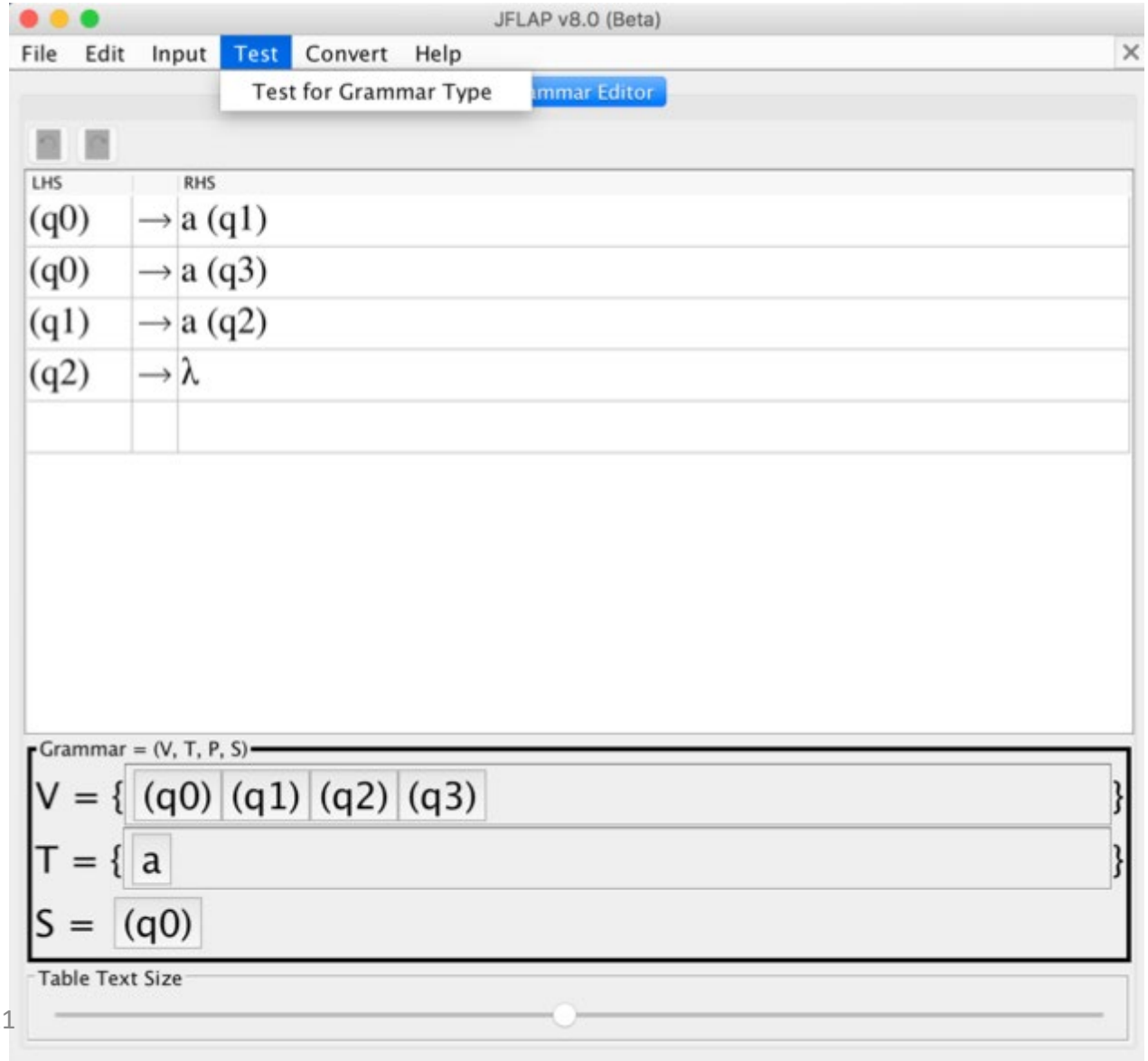
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Convert to Grammar
Step
Step To Completion
What's Left?
Export
q0
q1
q2
q3
a
a
LHS
RHS
(q0) → a (q1)
(q1) → a (q2)
(q0) → a (q3)
Table Text Size

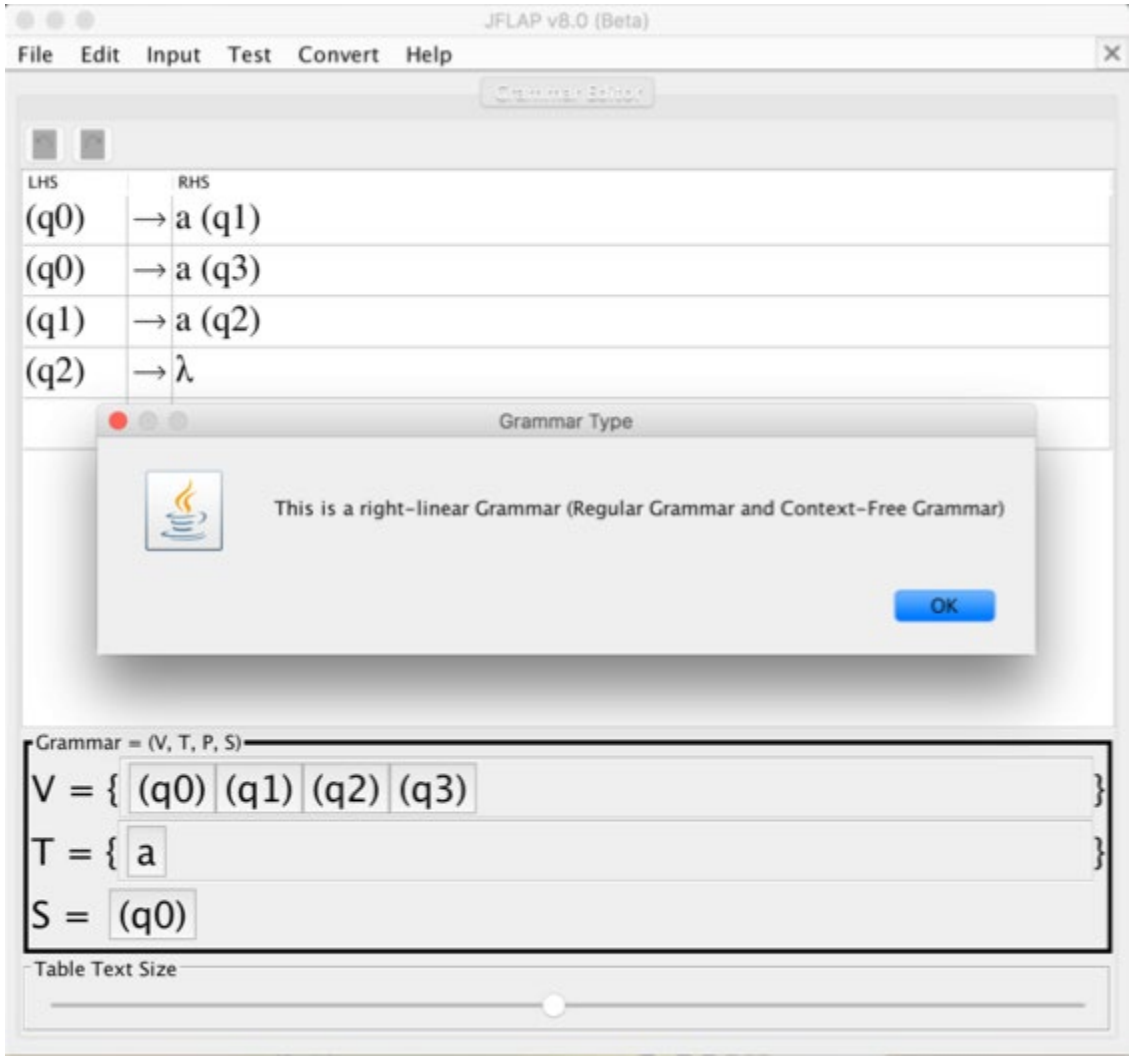
JFLAP: NFA


JFLAP v8.0 (Beta)
File
Input
Help
Automaton Editor
Convert to Grammar
Step
Step To Completion
What's Left?
Export
q0
q1
q2
q3
a
a
a
LHS
RHS
(q0) → a (q1)
(q1) → a (q2)
(q0) → a (q3)
(q2) → λ
Table Text Size

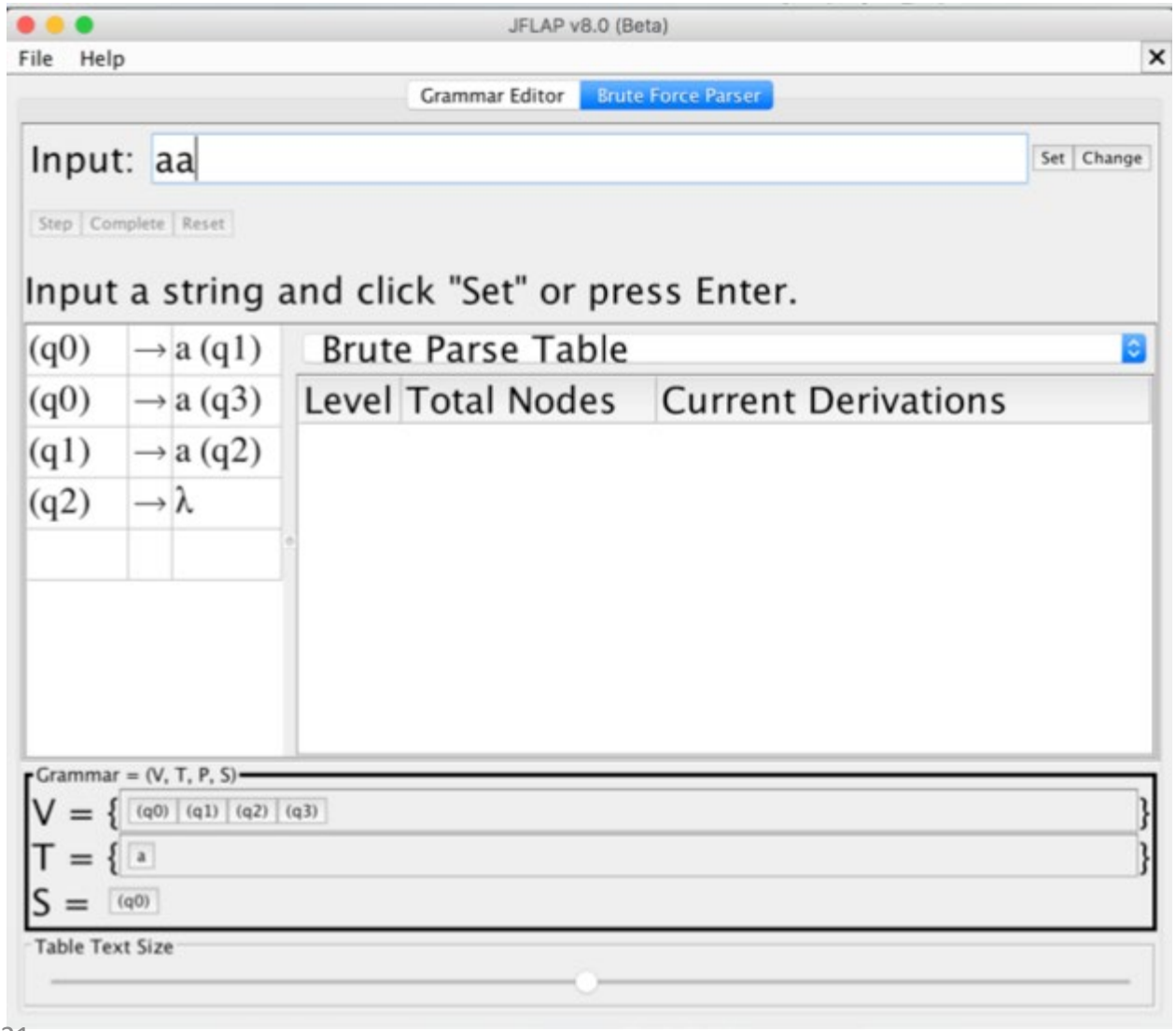
Di che tipo è la grammatica che genera L?



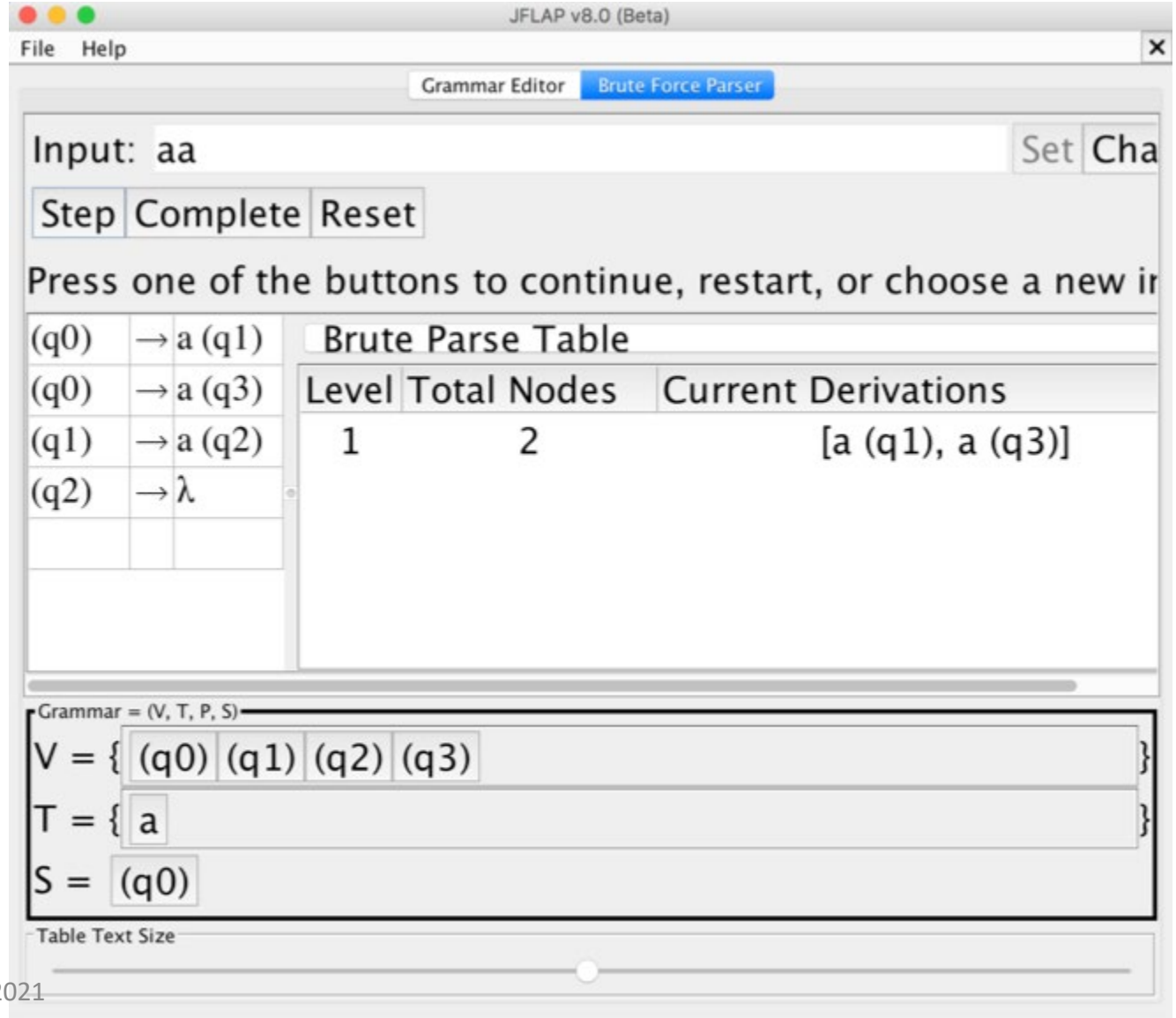
Di che tipo è la grammatica che genera L?



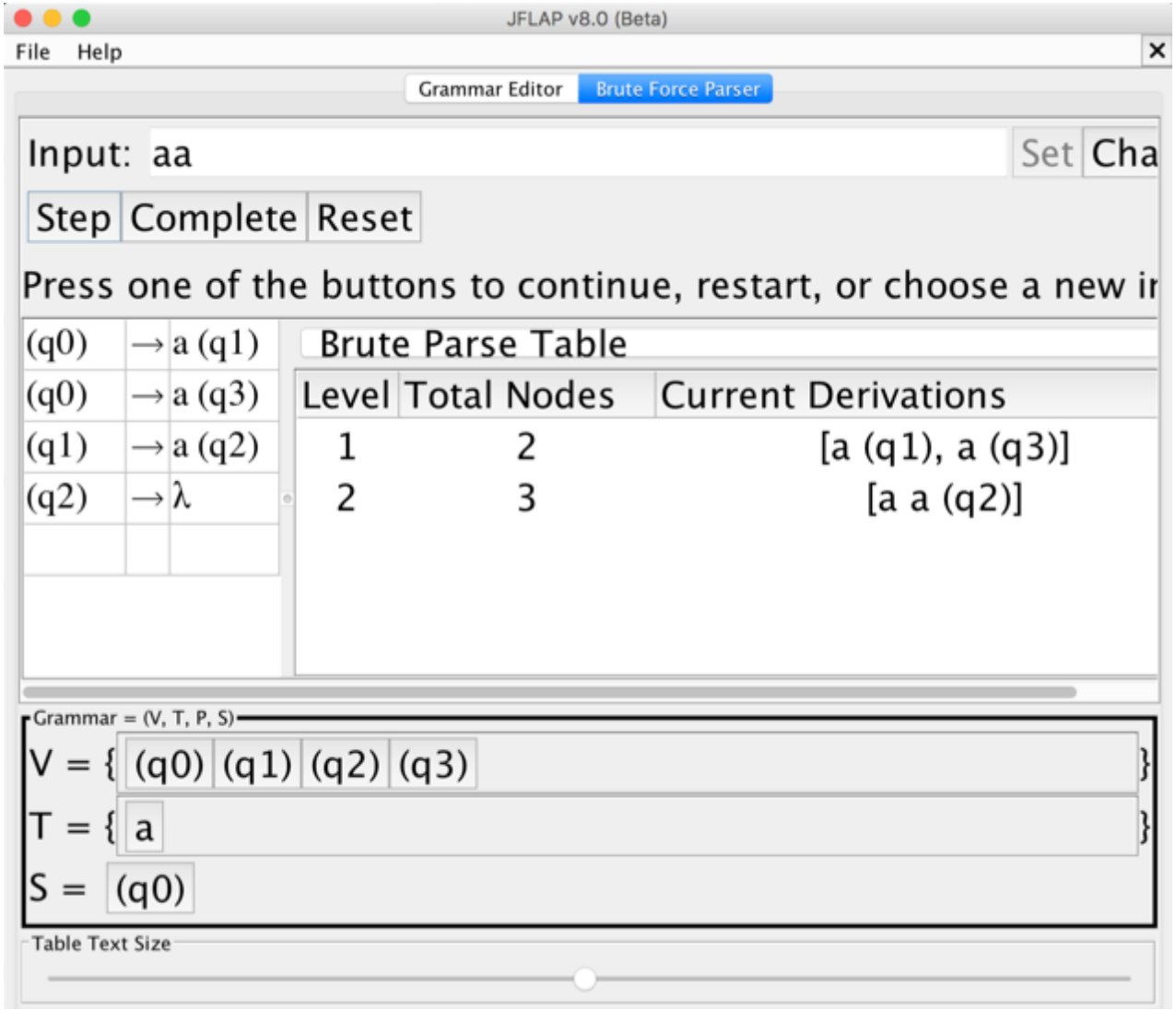
JFLAP: Brute Parse Table



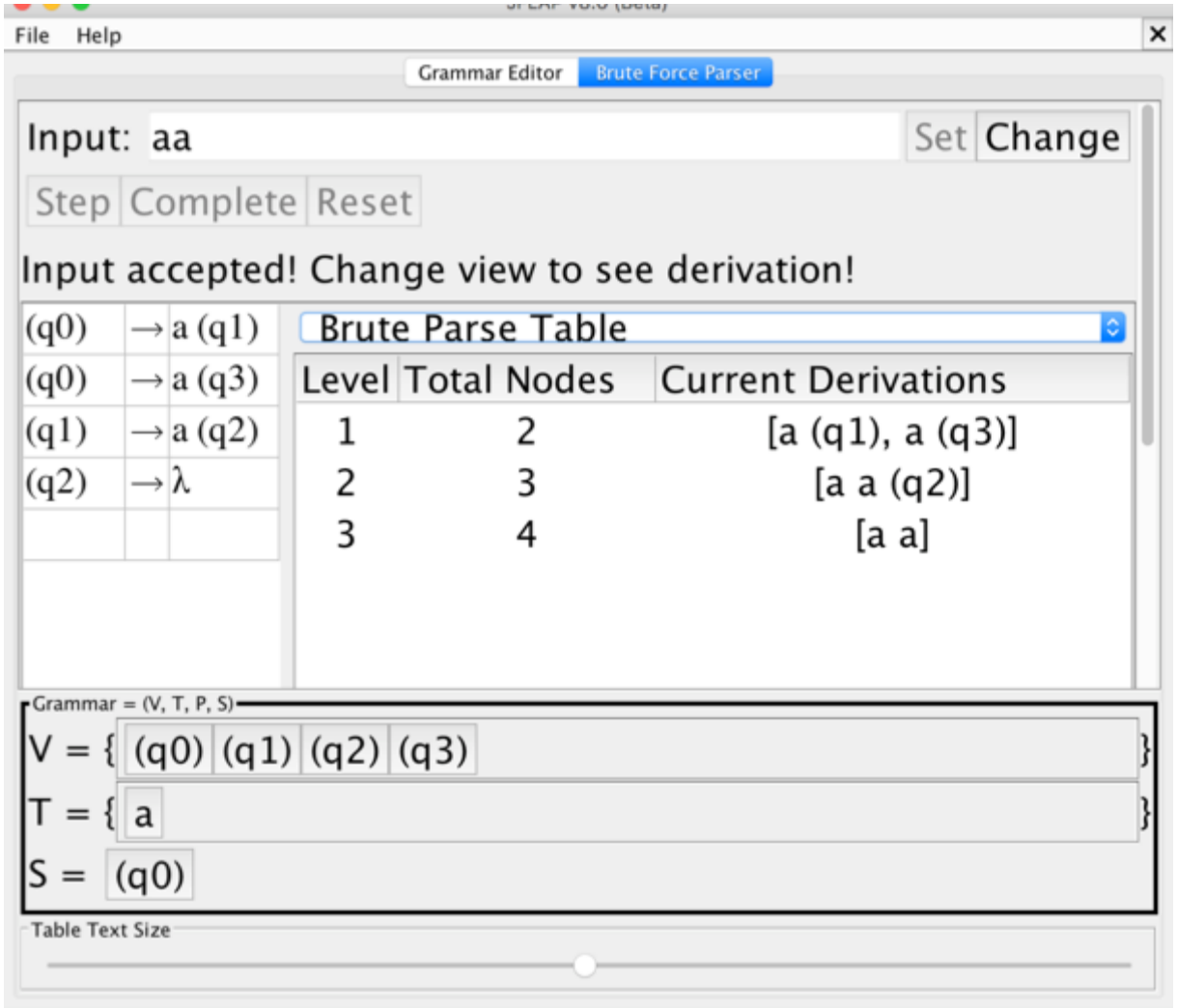
JFLAP: Brute Parse Table



JFLAP: Brute Parse Table



JFLAP: Brute Parse Table



JFLAP: Brute Parse Table

JFLAP v8.0 (Beta)

File Help

Grammar Editor | Brute Force Parser

Input: aaa Set Change

Step Complete Reset

Input rejected! Try another string!

(q0)	→	a (q1)
(q0)	→	a (q3)
(q1)	→	a (q2)
(q2)	→	λ

Brute Parse Table

Level	Total Nodes	Current Derivations
1	2	[a (q1), a (q3)]
2	3	[a a (q2)]
3	4	[a a]
4	4	[]

Grammar = (V, T, P, S)

V = { (q0) (q1) (q2) (q3) }

T = { a }

S = (q0)

Table Text Size

JFLAP: esempio

Espressione

REGOLARE

Linguaggio regolare *L*

- Quando abbiamo un linguaggio regolare *L* equivale:

Linguaggio *L* è in una rappresentazione standard

(DFA, NFA, or Regular Expression)

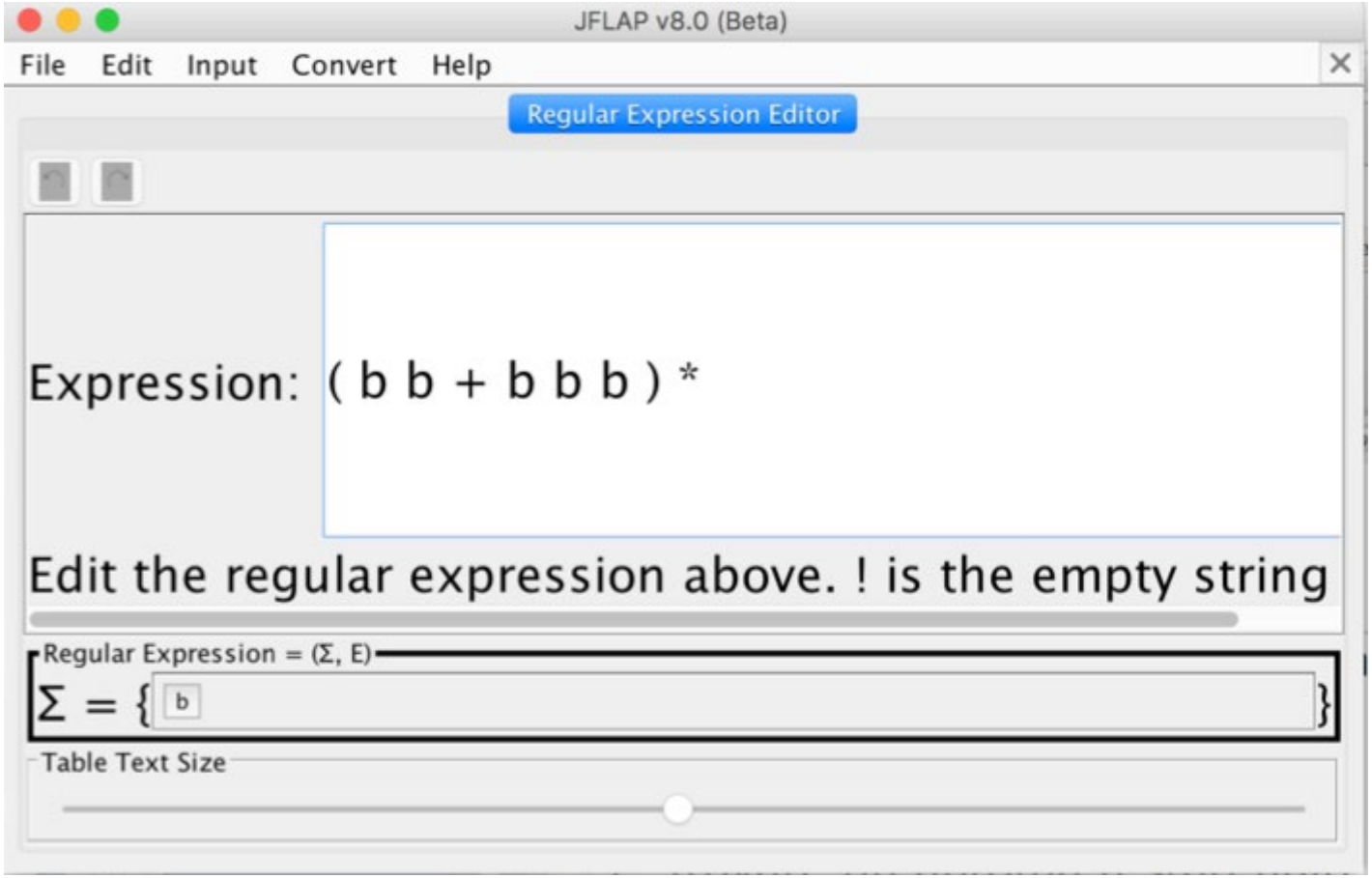
JFLAP: esercizio

- Sia *L* il linguaggio denotato dalla seguente espressione regolare:

$$(bb+bbb)^*$$

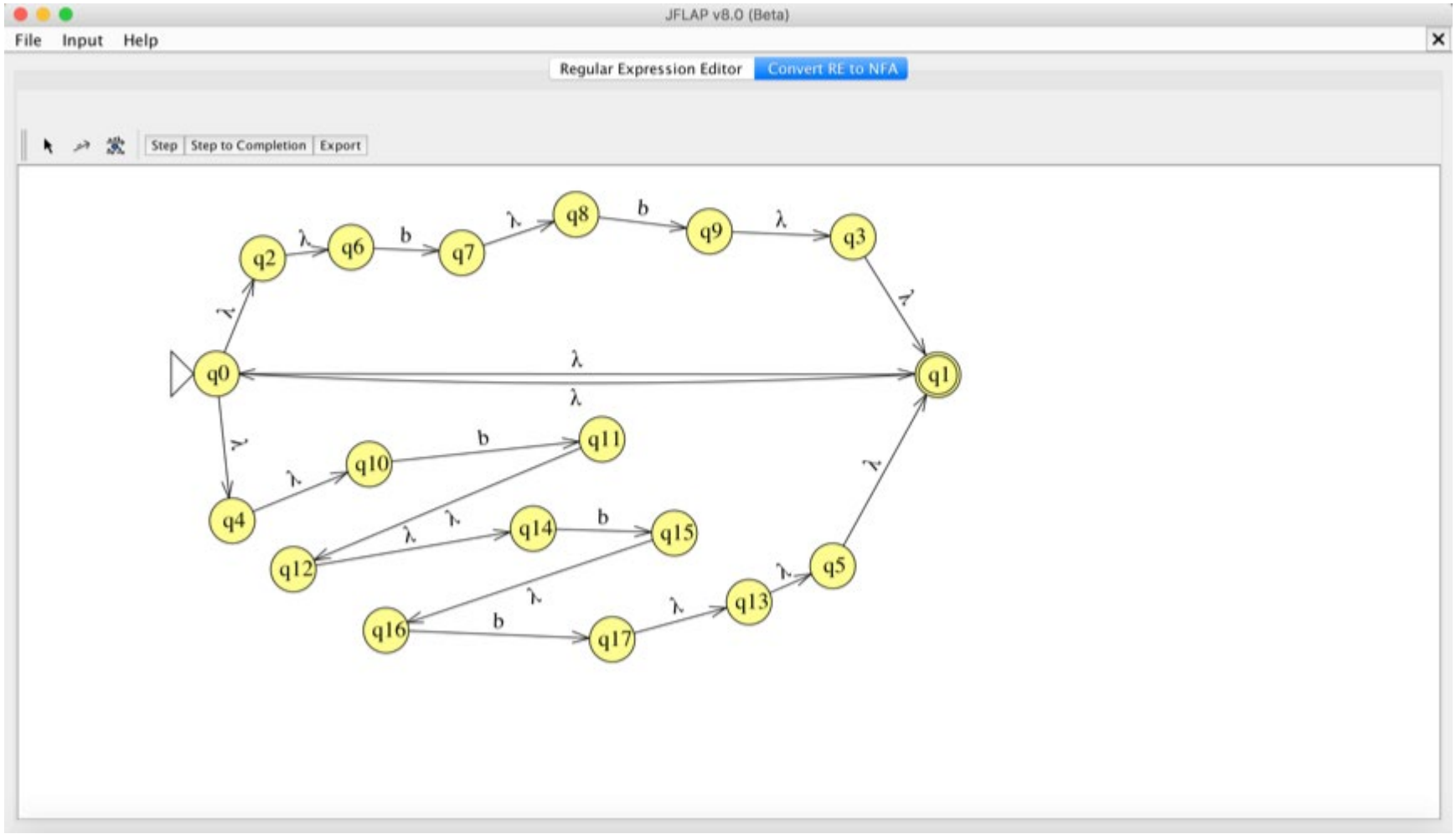
1. Trovare un automa a stati finiti che riconosce *L*
2. Trovare l'automa non deterministico trovato al punto 1. in automa deterministico equivalente

JFLAP: Espressione Regolare

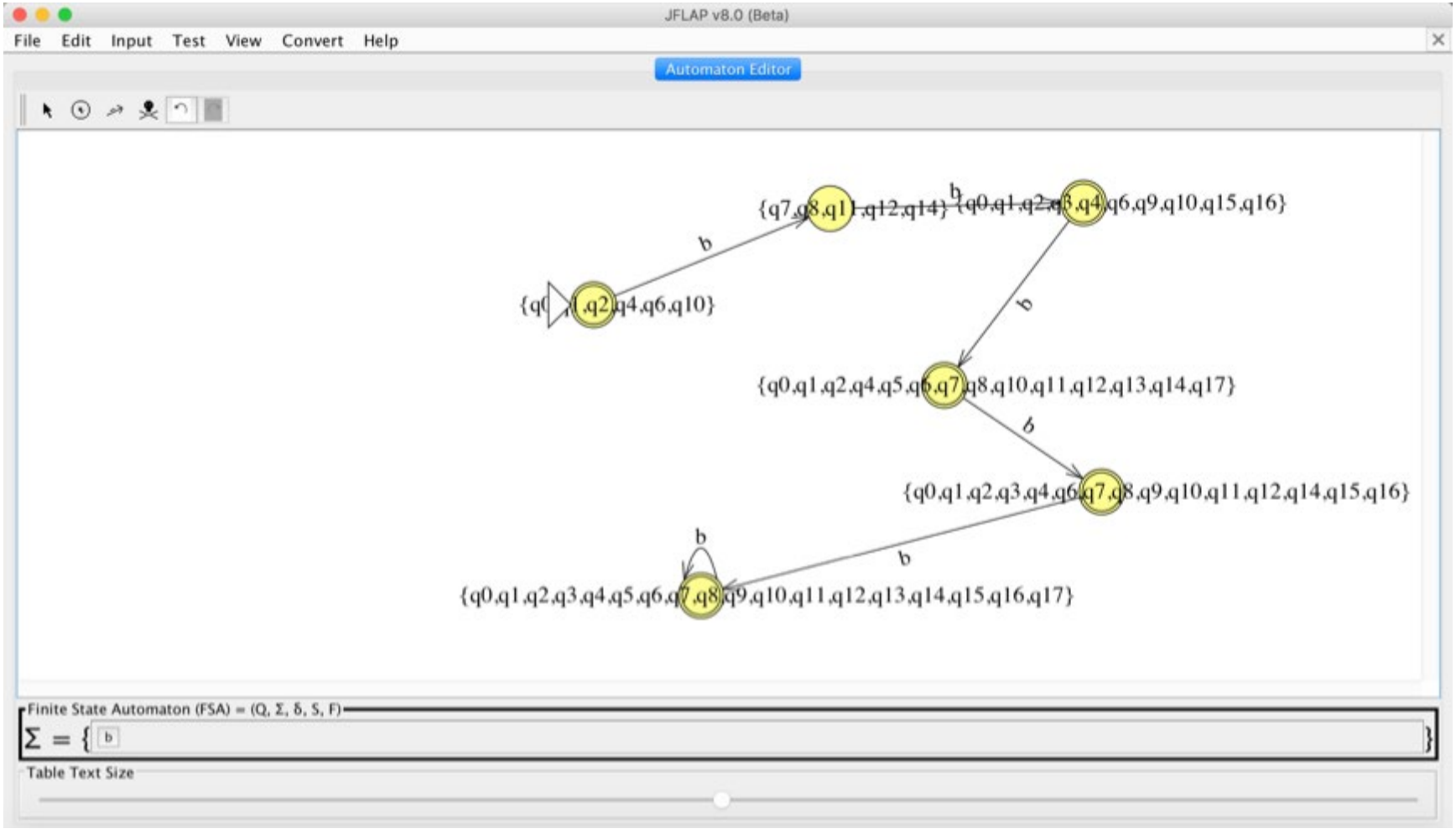


JFLAP: Espressione Regolare

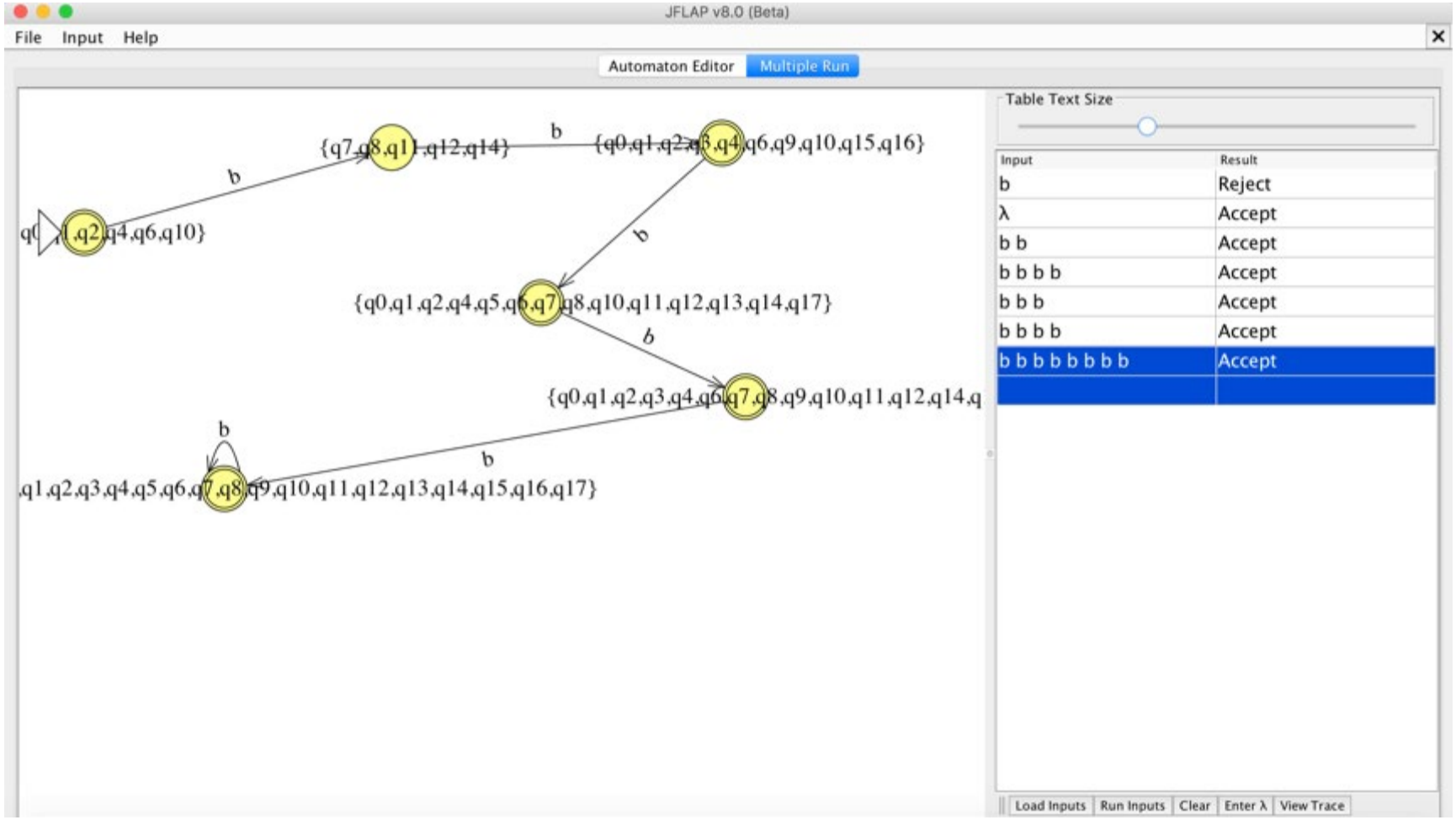
JFLAP: Espressione Regolare



JFLAP: Espressione Regolare



JFLAP: Espressione Regolare



JFLAP: esempio

Grammatiche

REGOLARI

Grammatiche regolari

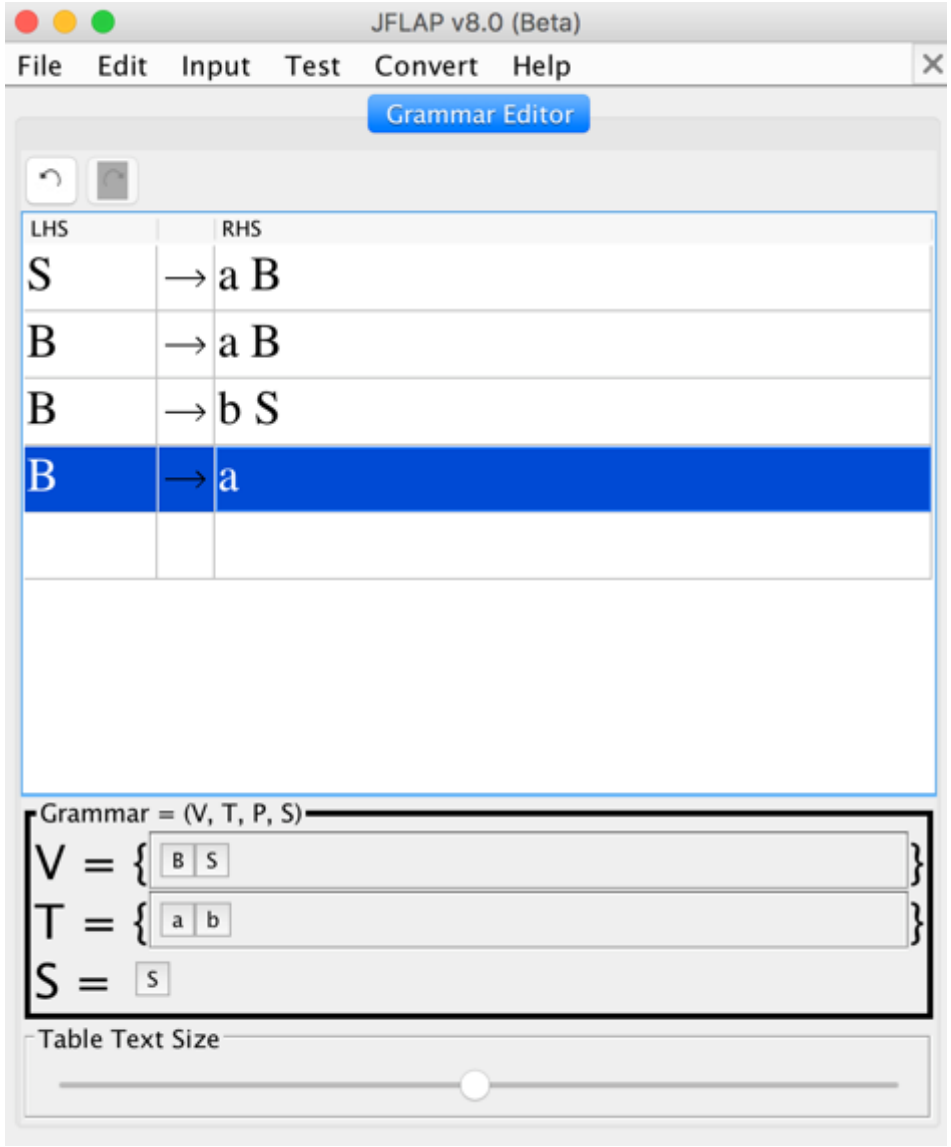
- Una grammatica regolare è una grammatica lineare destra o sinistra
- I linguaggi generati da una grammatica regolare è un linguaggio regolare

JFLAP: esercizio

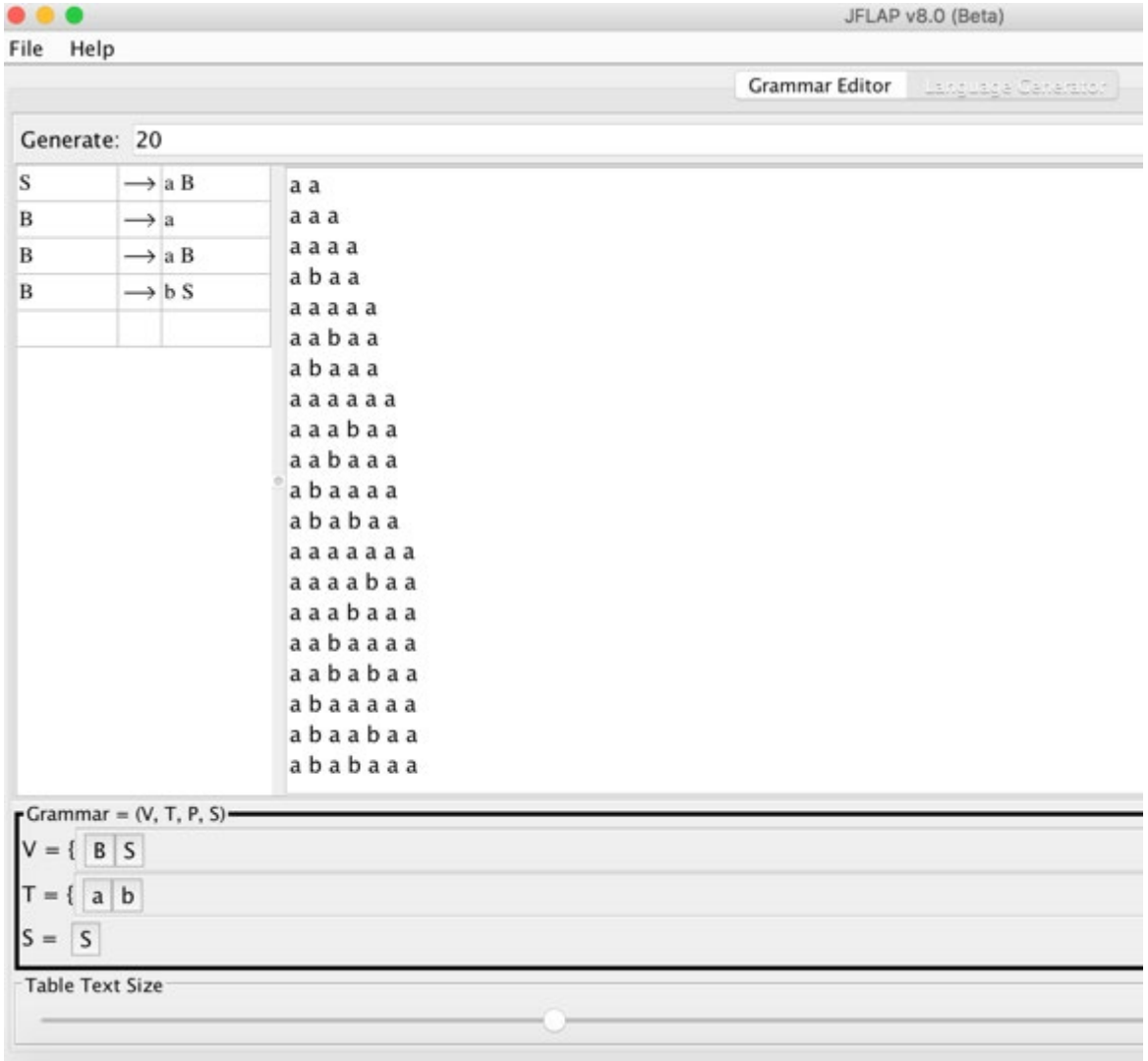
- Data la seguente grammatica lineare destra G=(X,V,S,P) con
  - X={a,b}
  - V={S,B}
  - P={S→aB,
    B→aB|bS|a}

1. Determinare un automa deterministico M tale che L(G)=T(M)

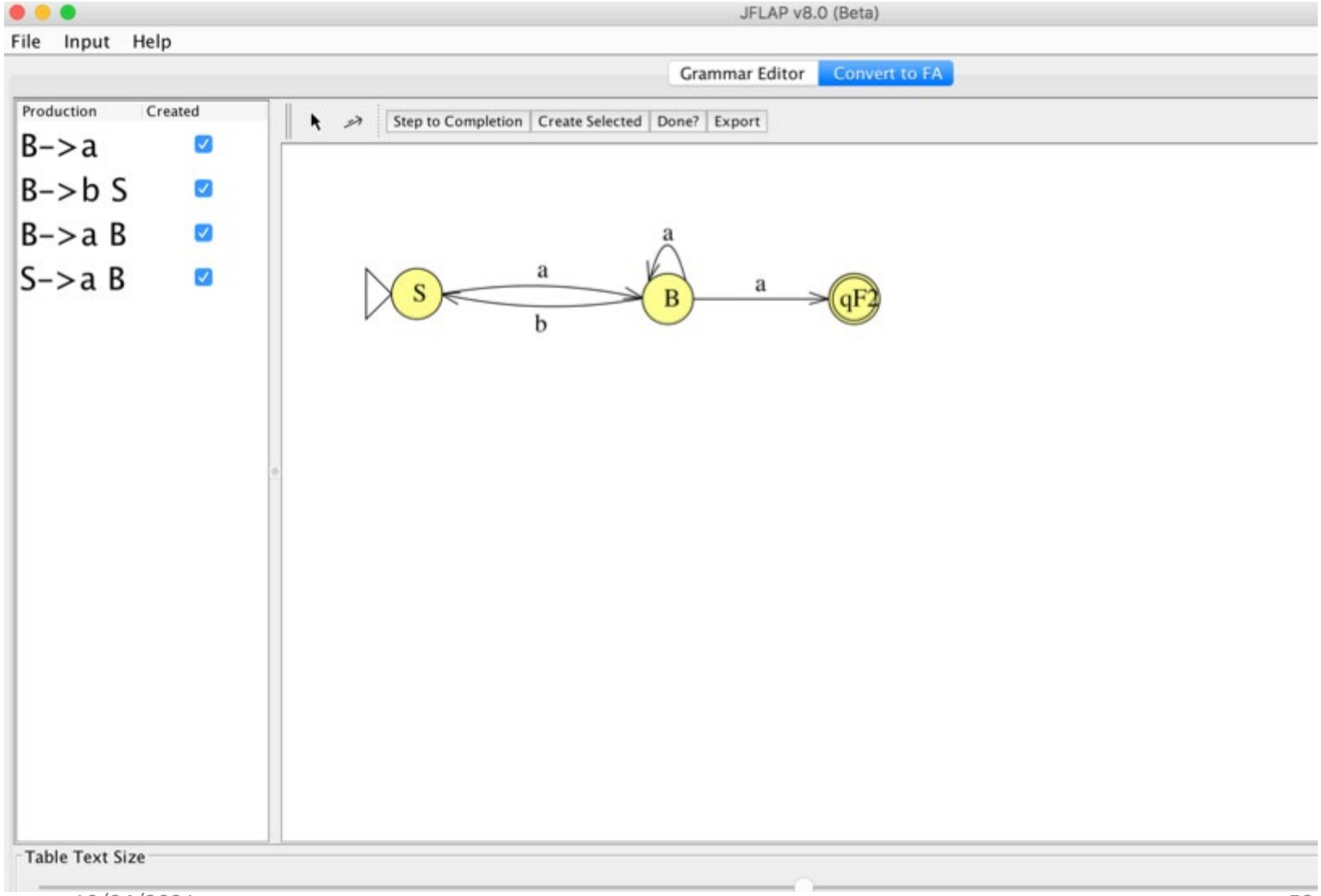
JFLAP: Grammatica lineare destra



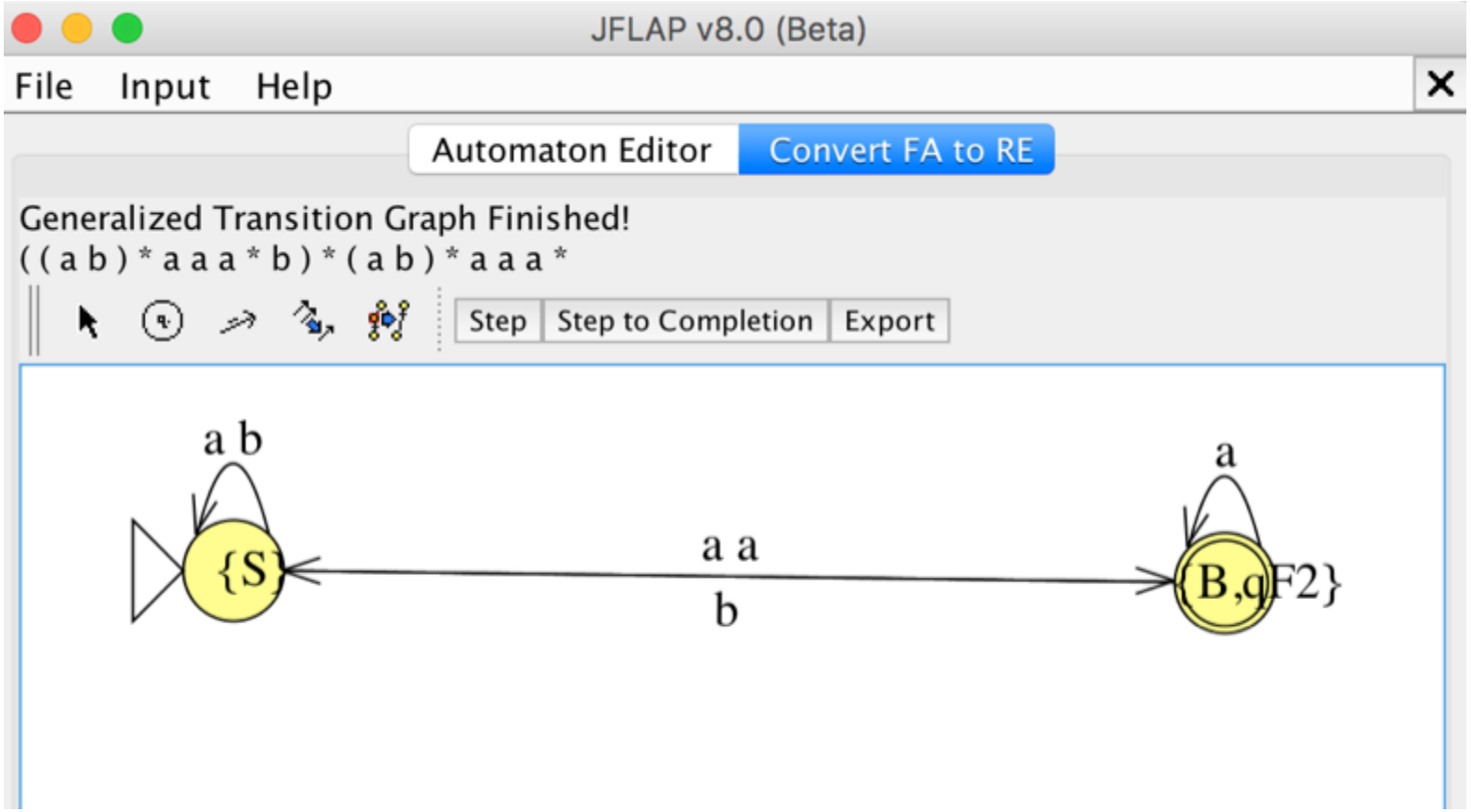
JFLAP: Grammatica lineare destra



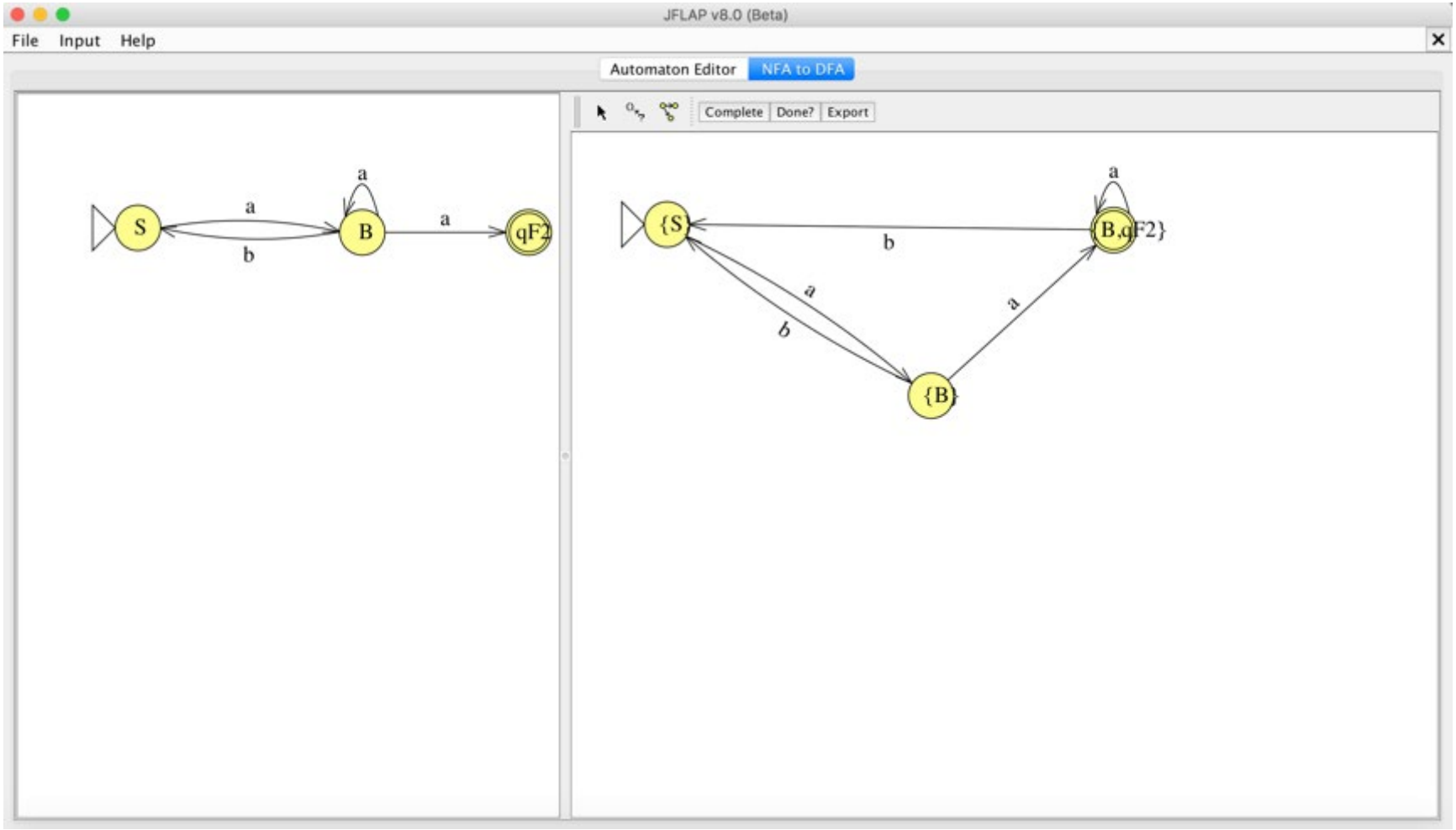
JFLAP: Grammatica lineare destra



JFLAP: Grammatica lineare destra



JFLAP: Grammatica lineare destra



Esercizio

- $L = \{ w \mid w \in \{a,b\}^*$, w ha un numero pari di a ed un numero dispari di b $\}$
  1. Costruire l'automa accettatore a stati finiti che riconosce L
  2. Determinare una grammatica lineare destra che genera il linguaggio