DFA Minimization

Module

Questions to Think About:

1. If all states are final, what is the language accepted by the DFA and the minimized DFA?

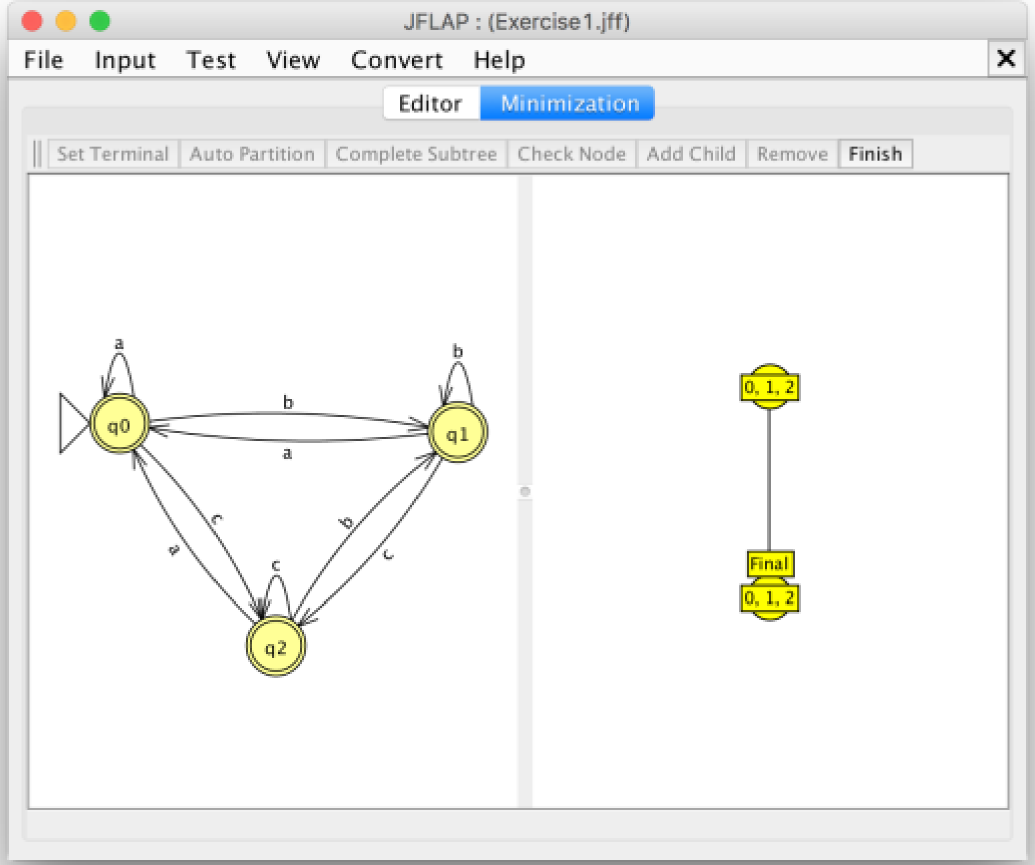
   Die Sprache besteht aus allen möglichen Konstellationen von Wörtern, die aus dem Alphabet der Sprache bestehen und dem Leerwort.

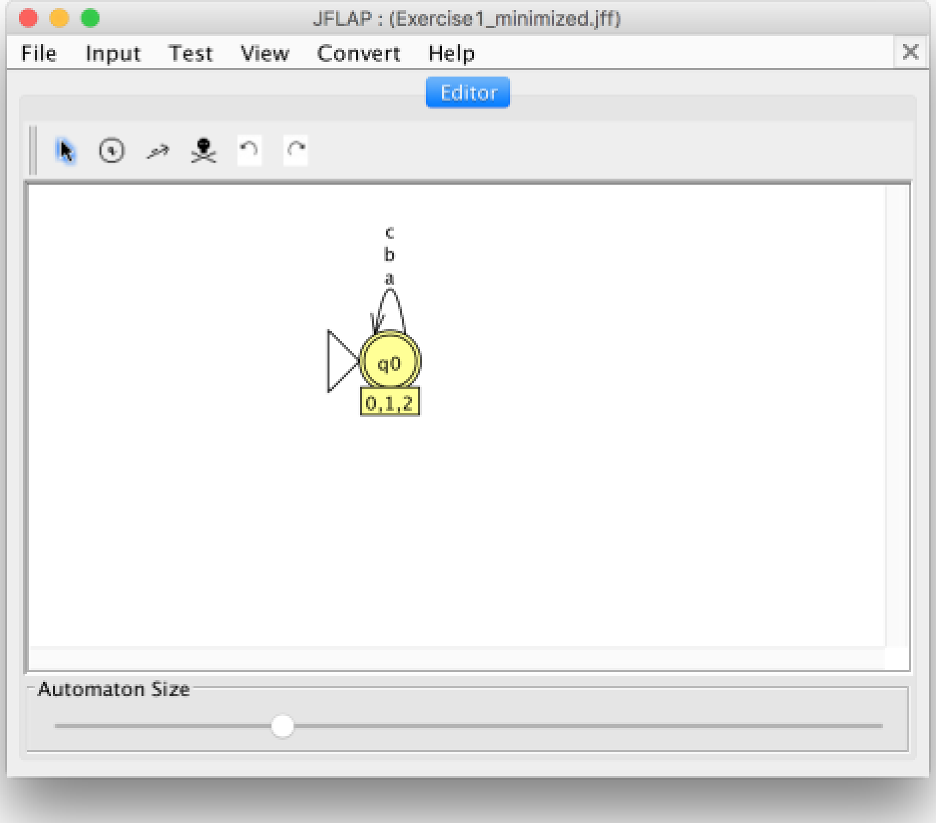
2. If all states are nonfinal, what is the language accepted by the DFA and the minimized DFA?

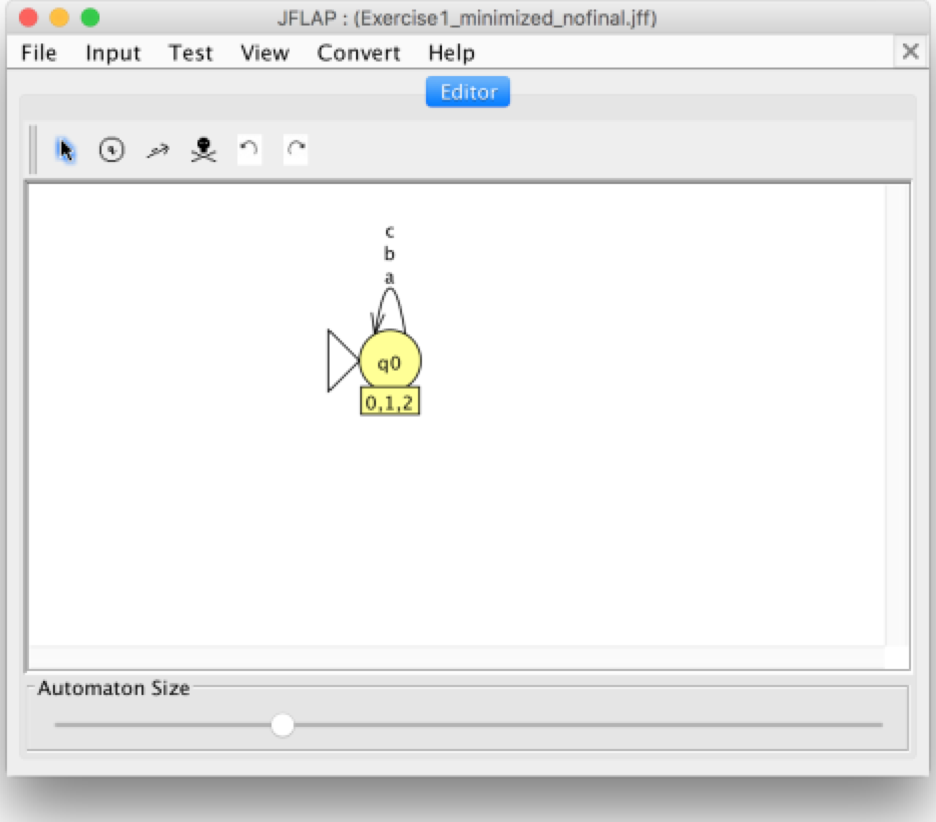
   Die Sprache ist eine leere Menge, da nie ein Endzustand erreicht wird.

3. Remember that Q represents the set of states in the DFA and that F represents the set of final states in the DFA. How can you write the set of nonfinal states using Q, F, and a set operation?

   Q / F


JFLAP : (Exercise1.jff)
File
Input
Test
View
Convert
Help
Editor
Minimization
Set Terminal
Auto Partition
Complete Subtree
Check Node
Add Child
Remove
Finish
a
b
b
a
q0
q1
c
a
b
c
c
q2
0, 1, 2
Final
0, 1, 2


JFLAP : (Exercise1_minimized.jff)
File
Input
Test
View
Convert
Help
Editor
c
b
a
q0
0,1,2
Automaton Size


JFLAP : (Exercise1_minimized_nofinal.jff)
File
Input
Test
View
Convert
Help
Editor
c
b
a
q0
0,1,2
Automaton Size

Questions to Think About

1. What language is accepted by the first minimized DFA? Use formal notation to describe it.

   $L_1 = \{a^j b^k c^l \mid j, k, l >= 0\}$

2. What language is accepted by the second minimized DFA? Use formal notation to describe it.

   $L_2 = \{\}$

3. What is the relationship between the previous two languages?

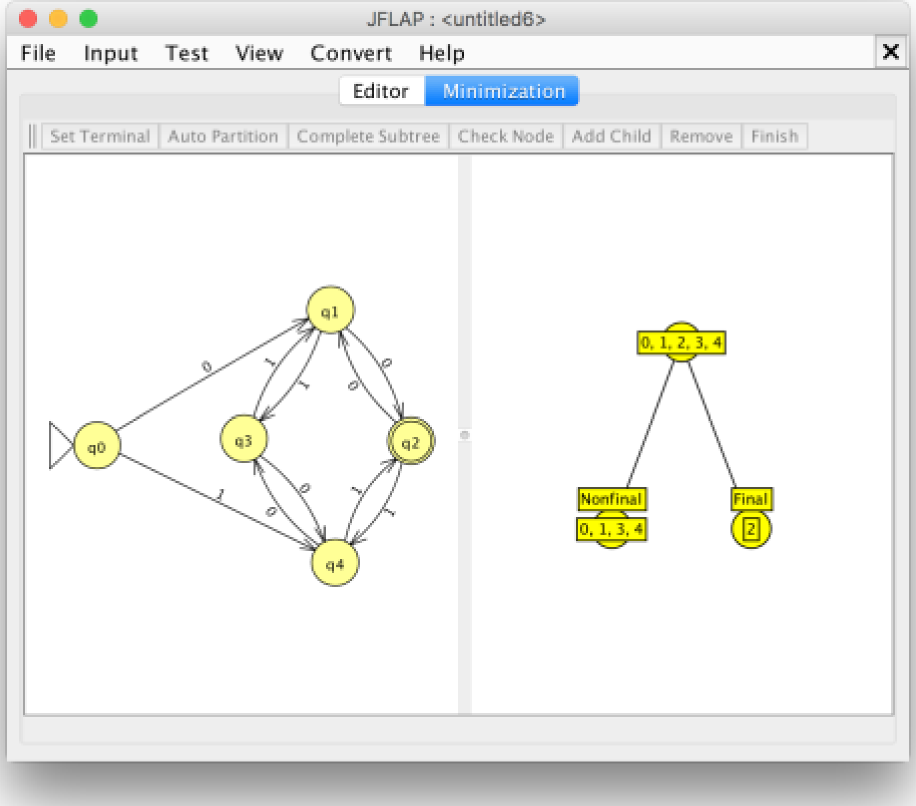
   L_2 ist das Komplement von L_1

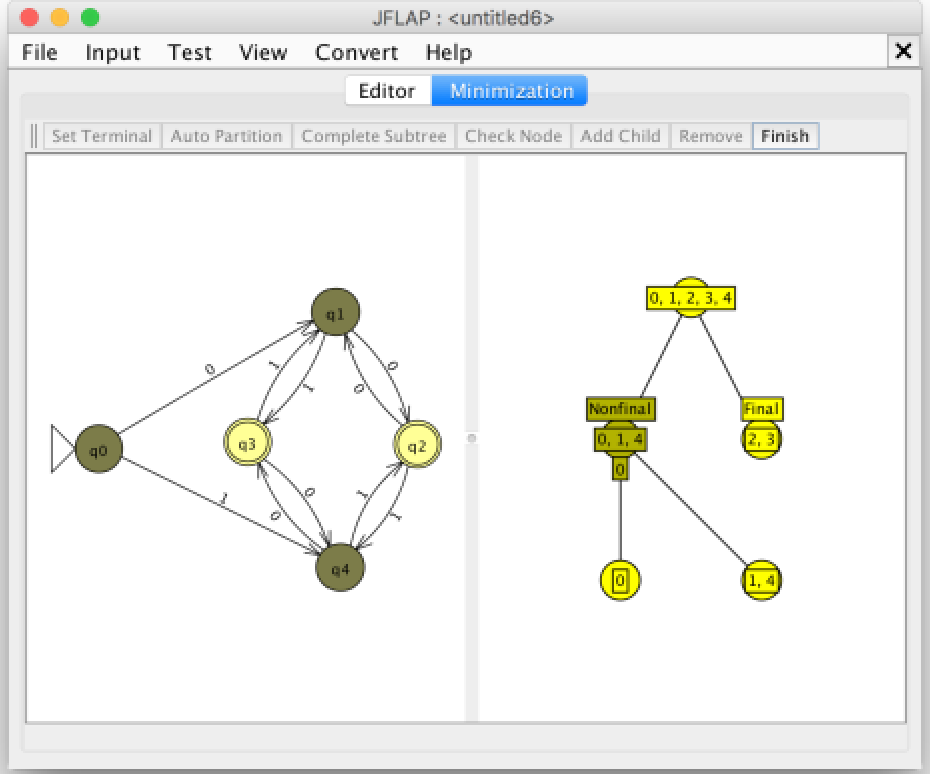
4. What happens to the language accepted by a DFA if you switch the roles of every state? That is, every final state becomes a nonfinal state, and vice versa.

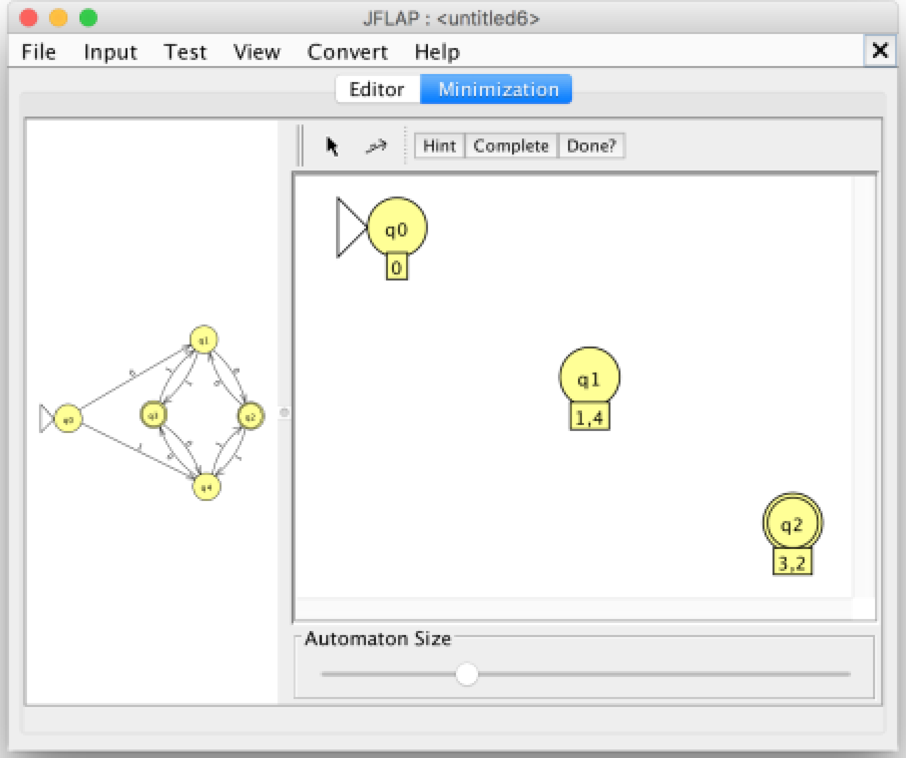
   Es entsteht die Komplementärsprache.

5. Can you predict what happens to the language accepted by an NFA if you switch the roles of every state? Why or why not?

   Nein, es entsteht eine komplett neue Art von Sprache.


JFLAP : <untitled6>
File
Input
Test
View
Convert
Help
Editor
Minimization
Set Terminal
Auto Partition
Complete Subtree
Check Node
Add Child
Remove
Finish
q0
q1
q2
q3
q4
0
1
0, 1, 2, 3, 4
Nonfinal
0, 1, 3, 4
Final
2


JFLAP : <untitled6>
File
Input
Test
View
Convert
Help
Editor
Minimization
Set Terminal
Auto Partition
Complete Subtree
Check Node
Add Child
Remove
Finish
q0
q1
q2
q3
q4
0
1
0, 1, 2, 3, 4
Nonfinal
0, 1, 4
0
Final
2, 3
0
1, 4



Questions to Think About

1. Which state should be the start state in the minimized DFA? Why?

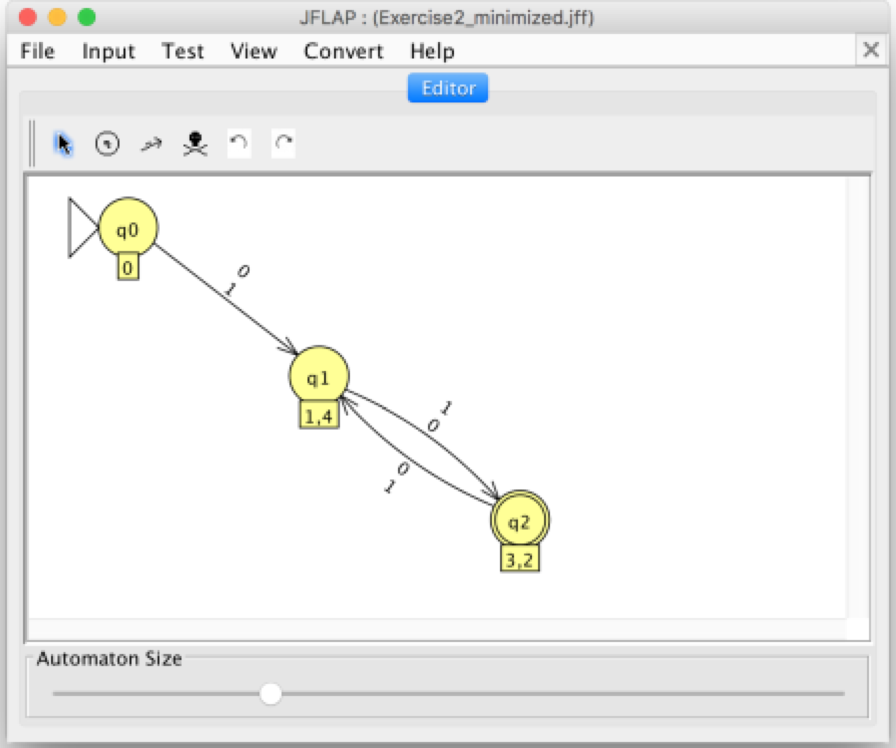
   q0, da er auch im originalen Automaten der Anfangs-Zustand ist.

2. Which states should be final states in the minimized DFA? Why?

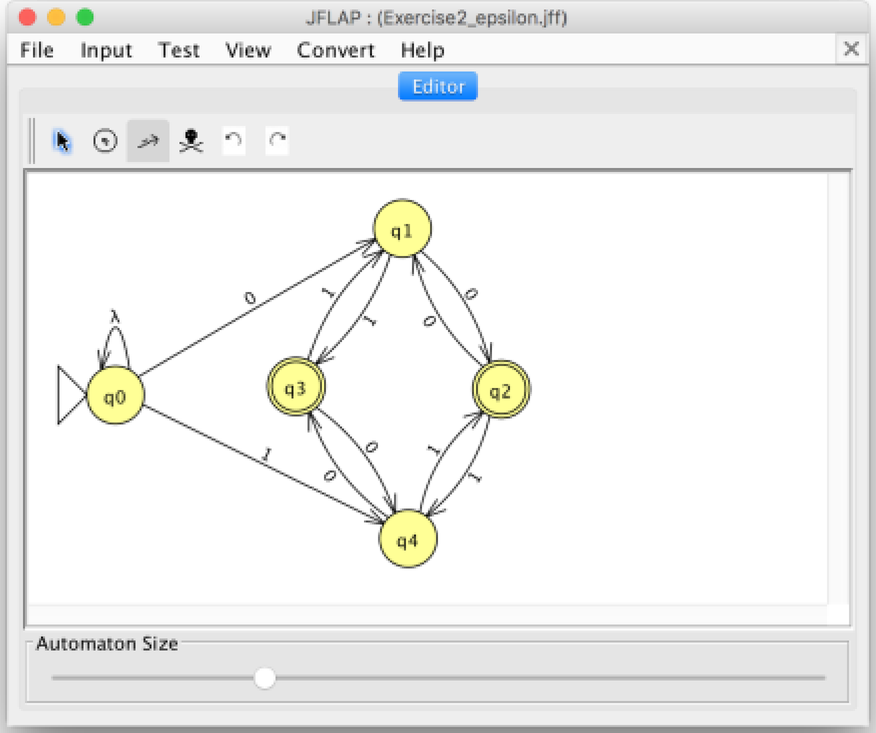
   q2, da dieser den Original-Zuständen q2 und q3 entspricht.

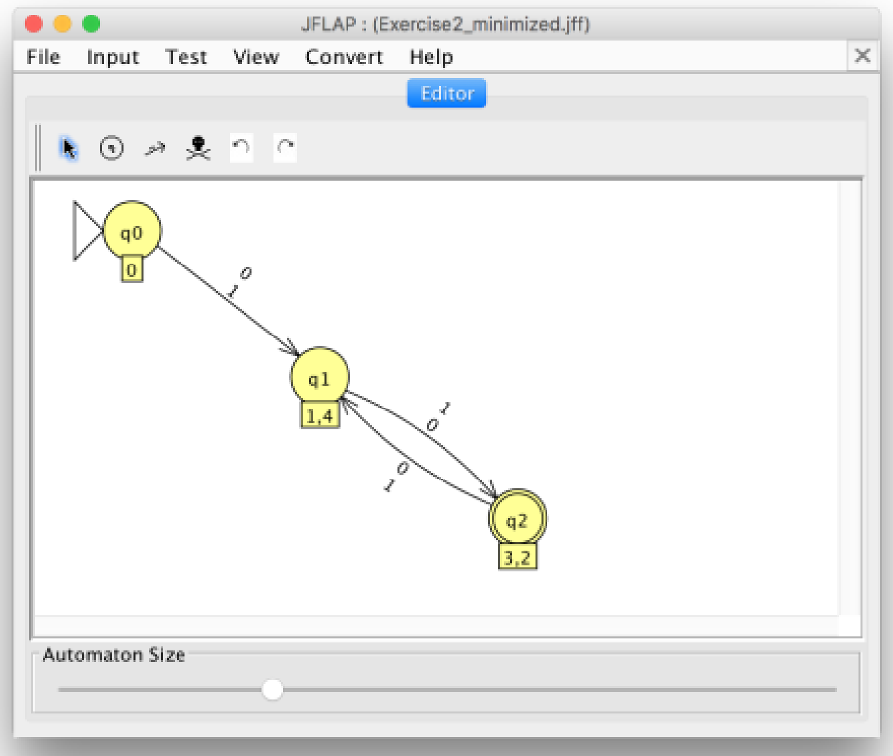
3. Can a final state ever be equivalent to a non-final state? Why or why not?

   Ja, wenn er dieselben Zeichen akzeptiert.


JFLAP : (Exercise2_minimized.jff)
File Input Test View Convert Help
Editor
q0
0
0
1
q1
1,4
1
0
0
1
q2
3,2
Automaton Size

Go back to the original DFA, and modify it so that it accepts the empty string because the empty string has 0 (an even number) occurrences of 0's and 0 occurences of 1's.


JFLAP : (Exercise2_epsilon.jff)
File Input Test View Convert Help
Editor
λ
q0
0
1
q1
1
1
0
0
q3
q2
0
0
1
1
q4
Automaton Size


JFLAP : (Exercise2_minimized.jff)
File Input Test View Convert Help
Editor
q0
0
0
1
q1
1,4
1
0
0
1
q2
3,2
Automaton Size

Questions to Think About

1. How many states are in the minimum-state DFA accepting all strings over alphabet Σ?

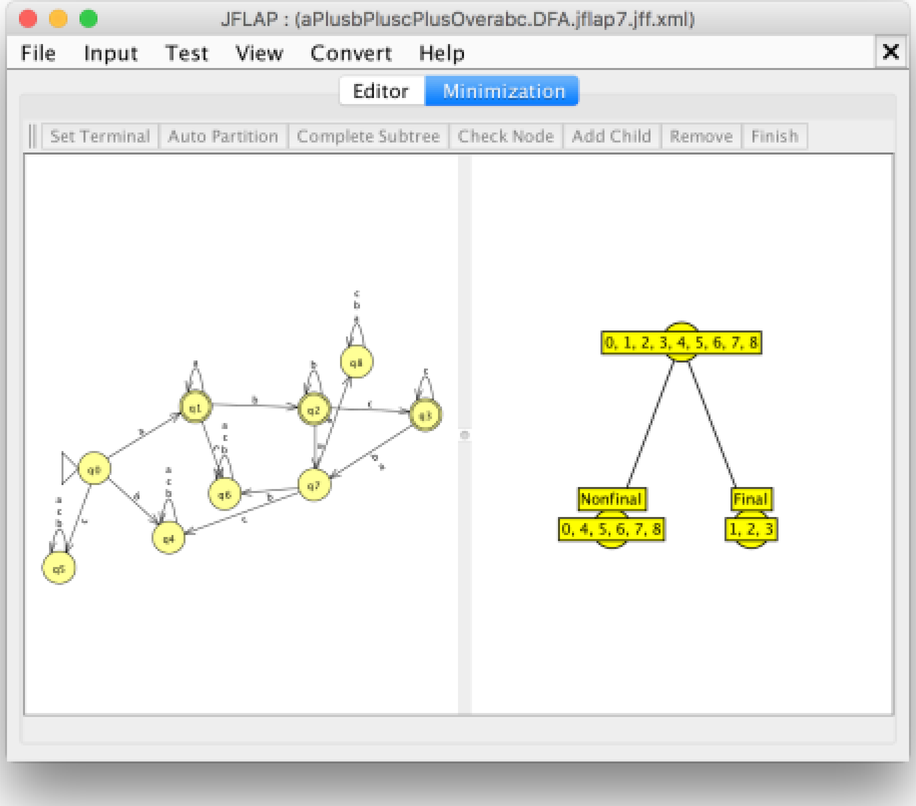
   2 (q1, q2)

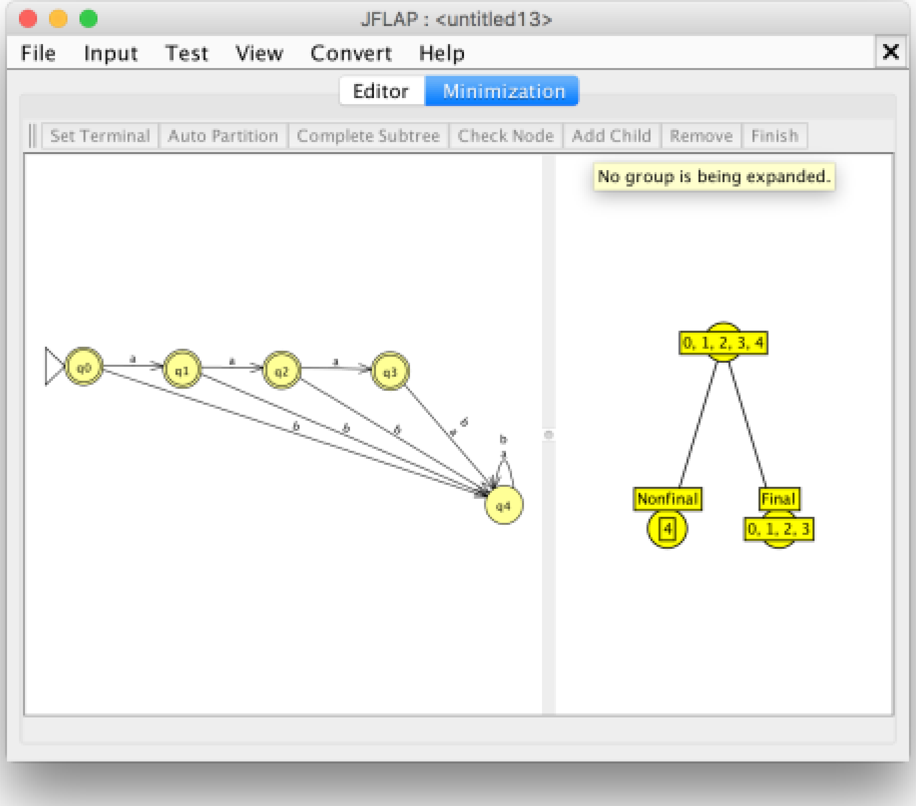
2. How many states are in the minimum-state DFA accepting no strings over alphabet Σ?

   0

3. How many states are in the minimum-state DFA accepting a single string of length n?

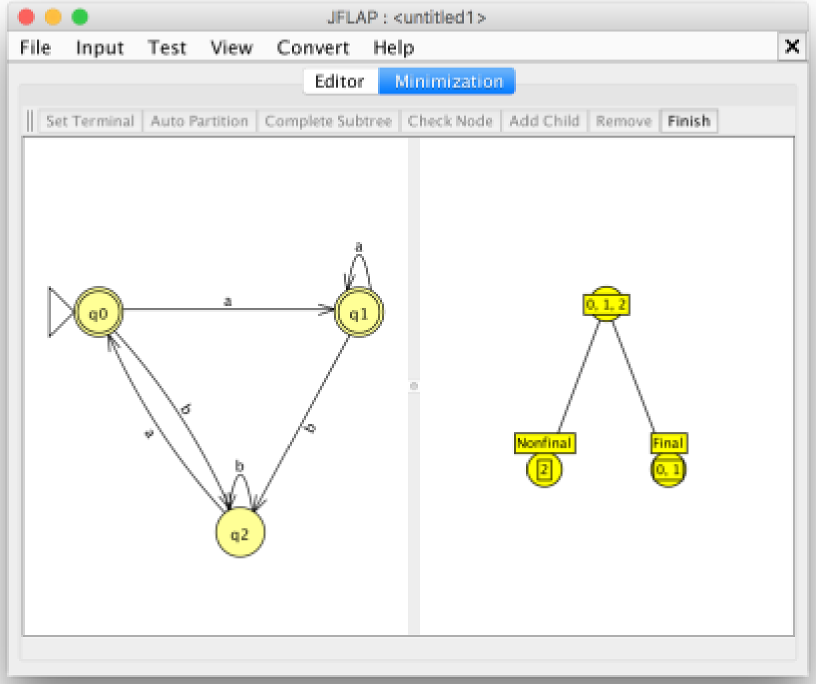
   1 (q0)

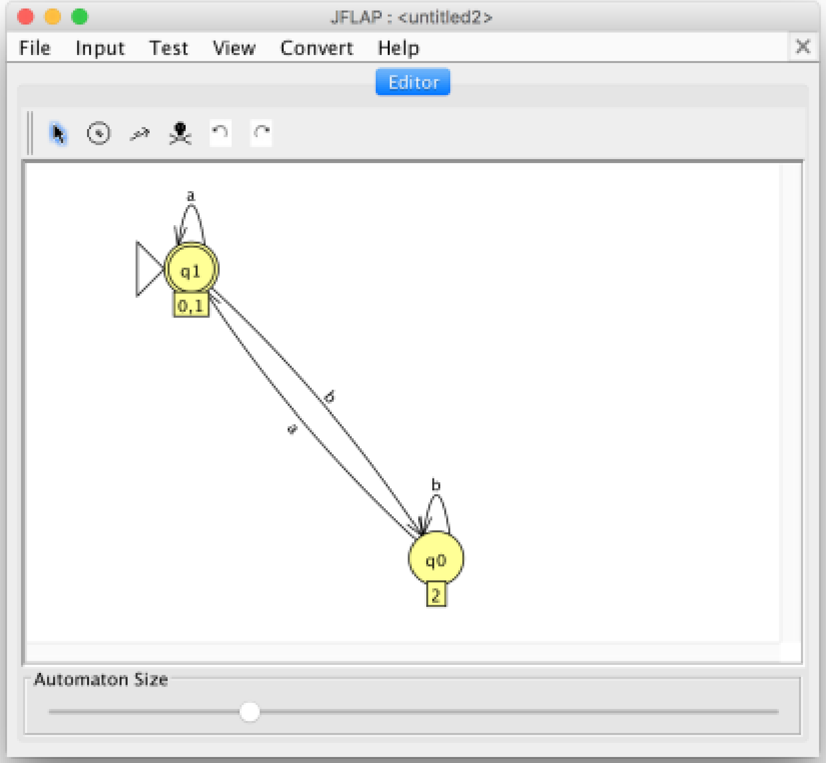


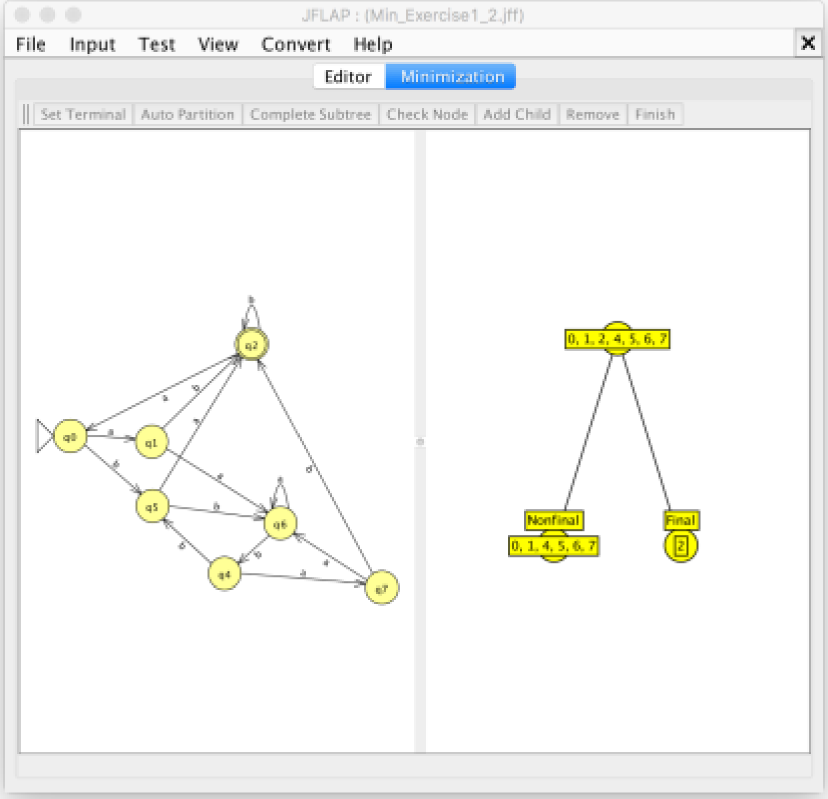


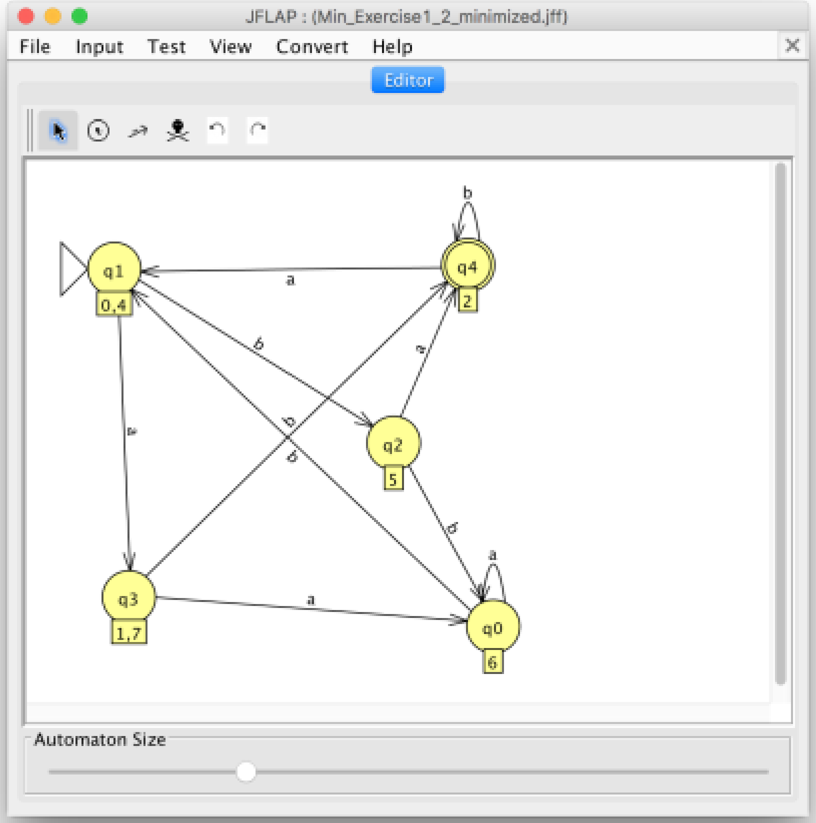
No group is being expanded.

Exercise 1

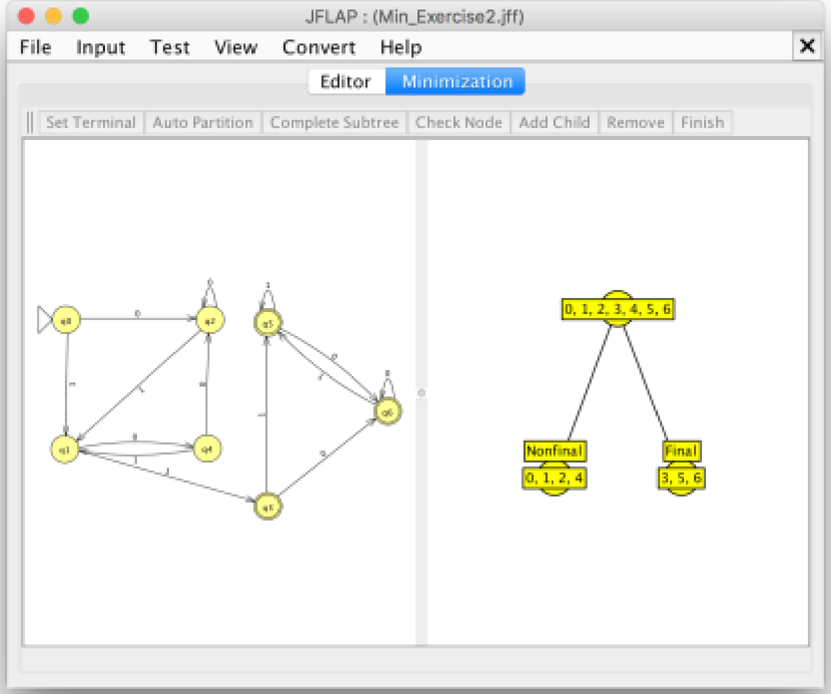

JFLAP : <untitled1>
File Input Test View Convert Help
Editor Minimization
Set Terminal Auto Partition Complete Subtree Check Node Add Child Remove Finish
q0
q1
q2
a
a
b
a
b
b
0, 1, 2
Nonfinal
2
Final
0, 1

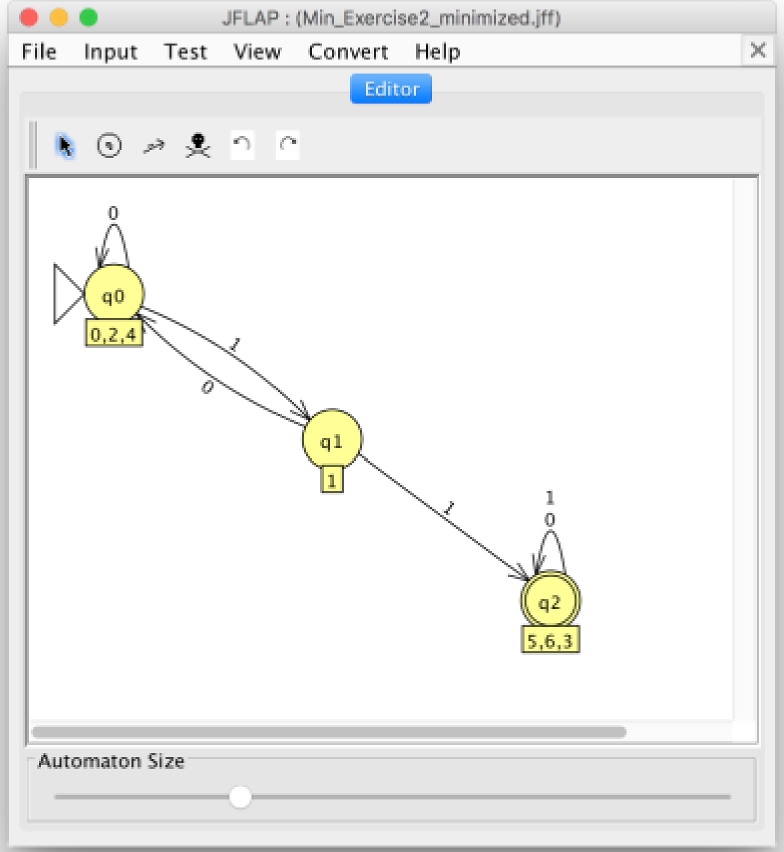

JFLAP : <untitled2>
File Input Test View Convert Help
Editor
q1
0,1
q0
2
a
b
a
b
Automaton Size


JFLAP : (Min_Exercise1_2.jff)
File Input Test View Convert Help
Editor Minimization
Set Terminal Auto Partition Complete Subtree Check Node Add Child Remove Finish
q0
q1
q2
q4
q5
q6
q7
0, 1, 2, 4, 5, 6, 7
Nonfinal
0, 1, 4, 5, 6, 7
Final
2

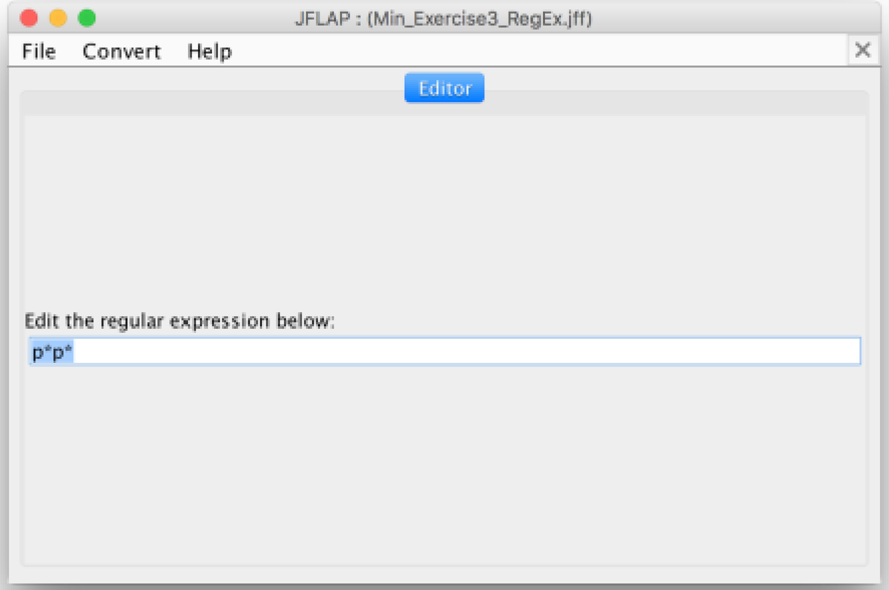

JFLAP : (Min_Exercise1_2_minimized.jff)
File Input Test View Convert Help
Editor
q1
0,4
q4
2
q2
5
q3
1,7
q0
6
a
b
Automaton Size

Exercise 2


JFLAP : (Min_Exercise2.jff)
File Input Test View Convert Help
Editor Minimization
Set Terminal Auto Partition Complete Subtree Check Node Add Child Remove Finish
0, 1, 2, 3, 4, 5, 6
Nonfinal
0, 1, 2, 4
Final
3, 5, 6

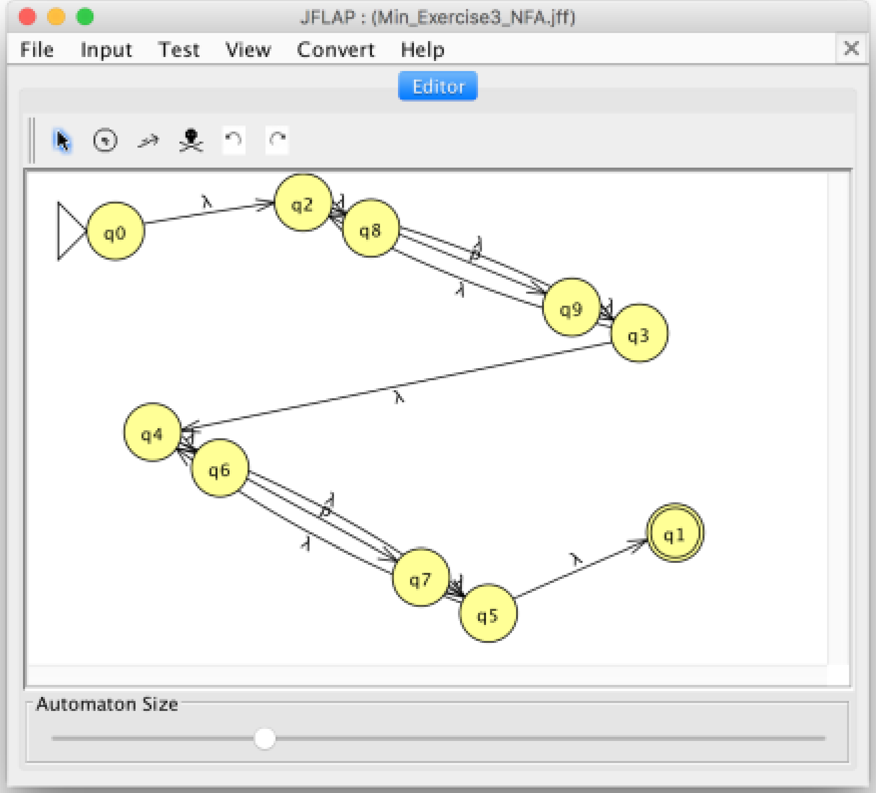

JFLAP : (Min_Exercise2_minimized.jff)
File Input Test View Convert Help
Editor
0
q0
0,2,4
1
0
q1
1
1
1
0
q2
5,6,3
Automaton Size

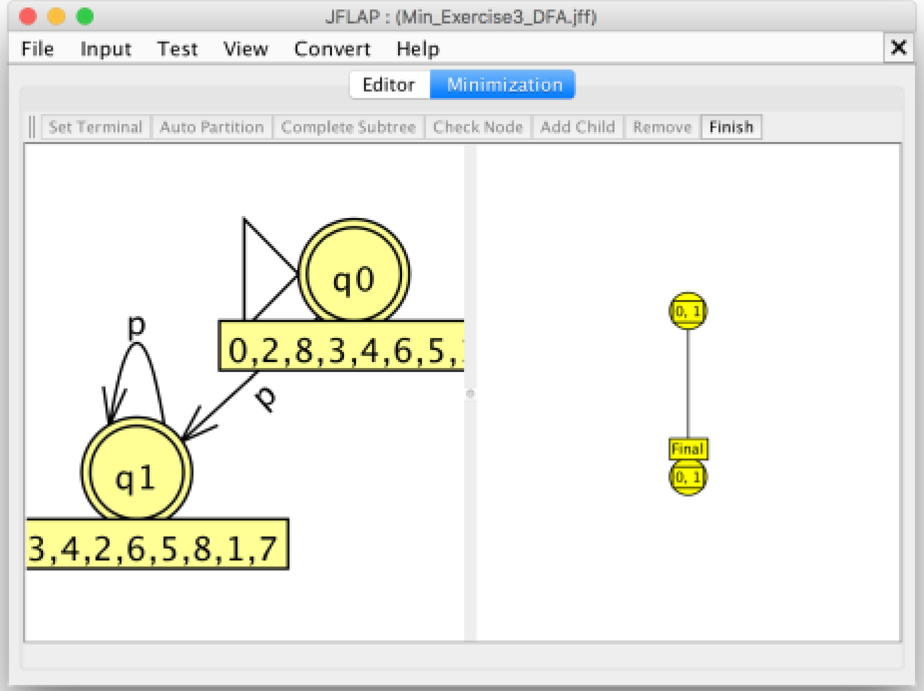
Exercise 3

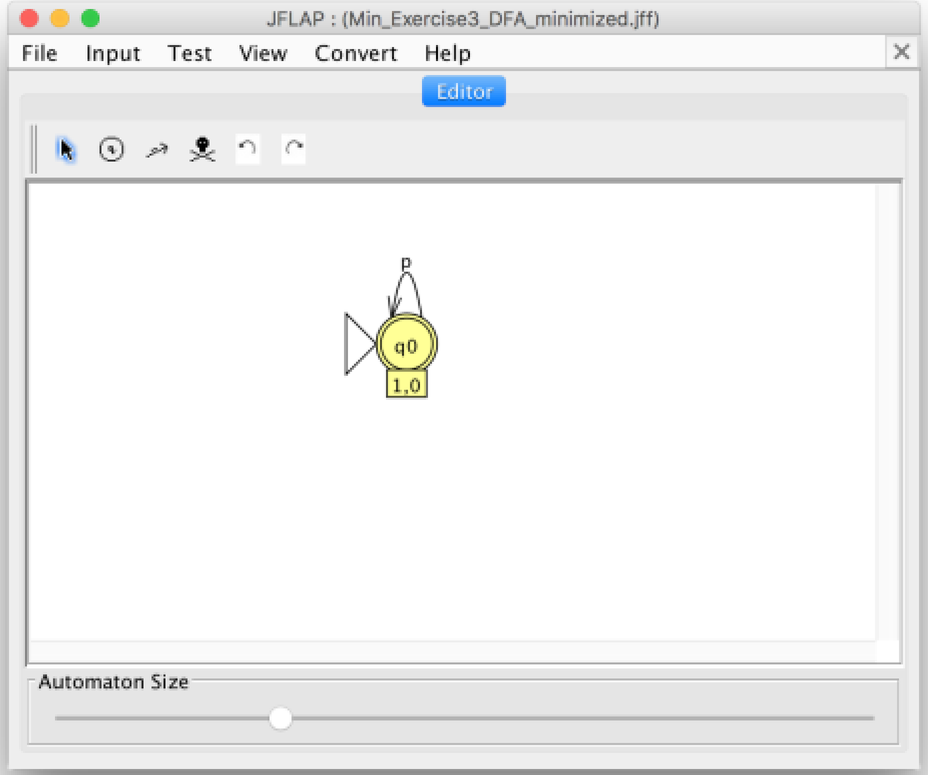


Edit the regular expression below:

p*p*

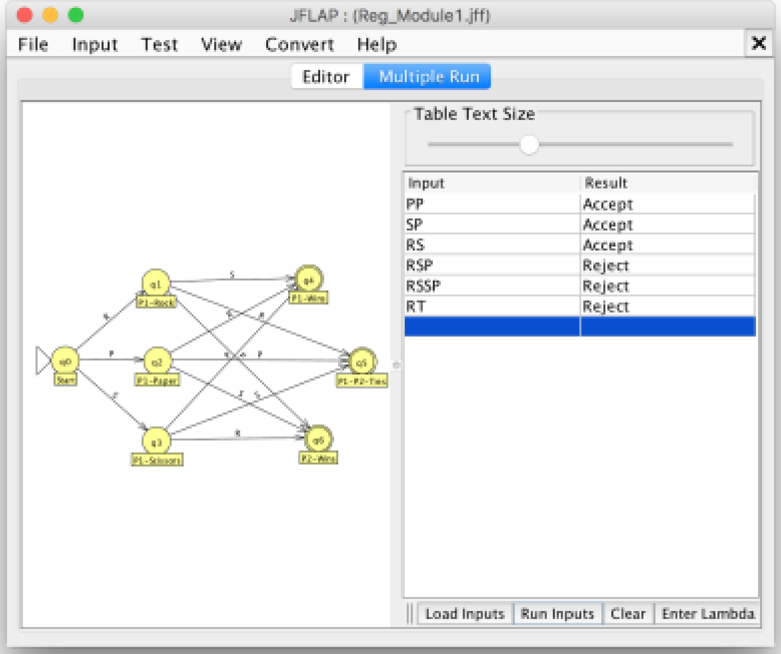



JFLAP : (Min_Exercise3_DFA.jff)
File Input Test View Convert Help
Editor Minimization
Set Terminal Auto Partition Complete Subtree Check Node Add Child Remove Finish
q0
0,2,8,3,4,6,5,
p
p
q1
3,4,2,6,5,8,1,7
0, 1
Final
0, 1

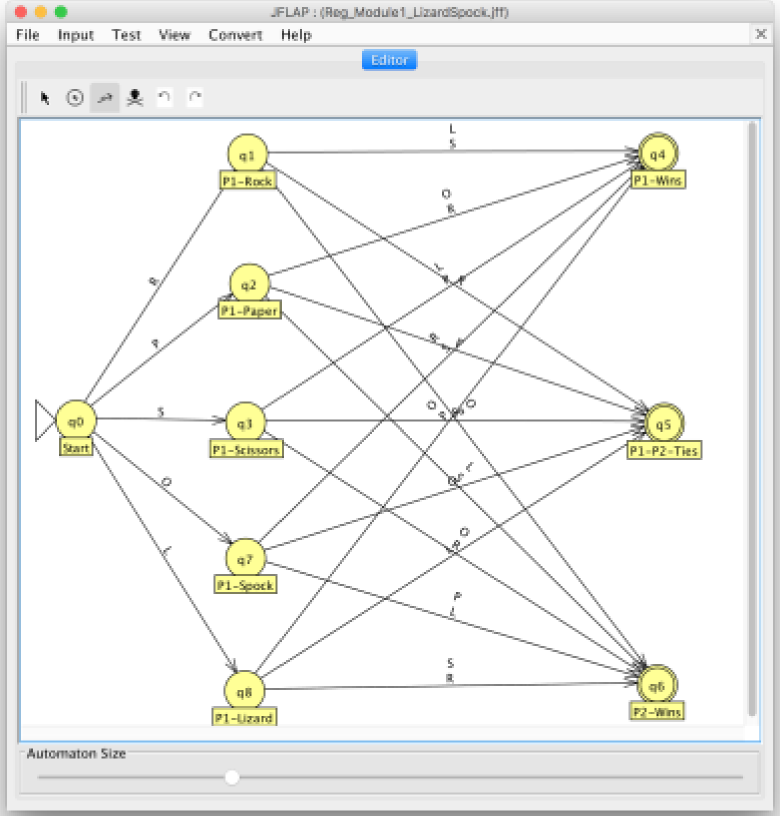

JFLAP : (Min_Exercise3_DFA_minimized.jff)
File Input Test View Convert Help
Editor
p
q0
1,0
Automaton Size

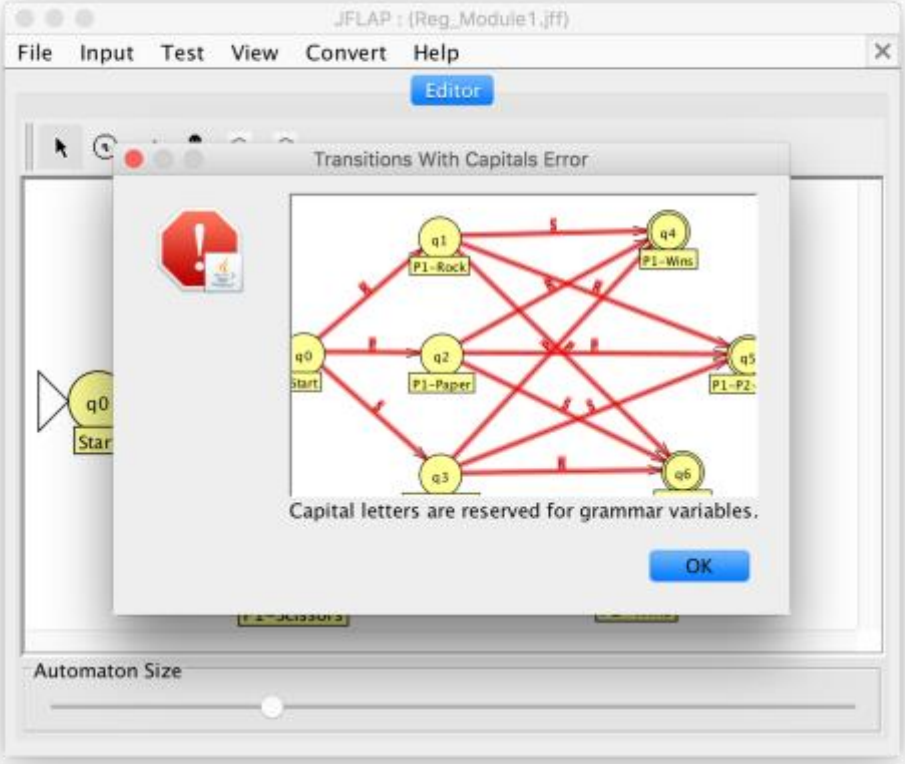
DFA to Regular Grammar Conversion

Module

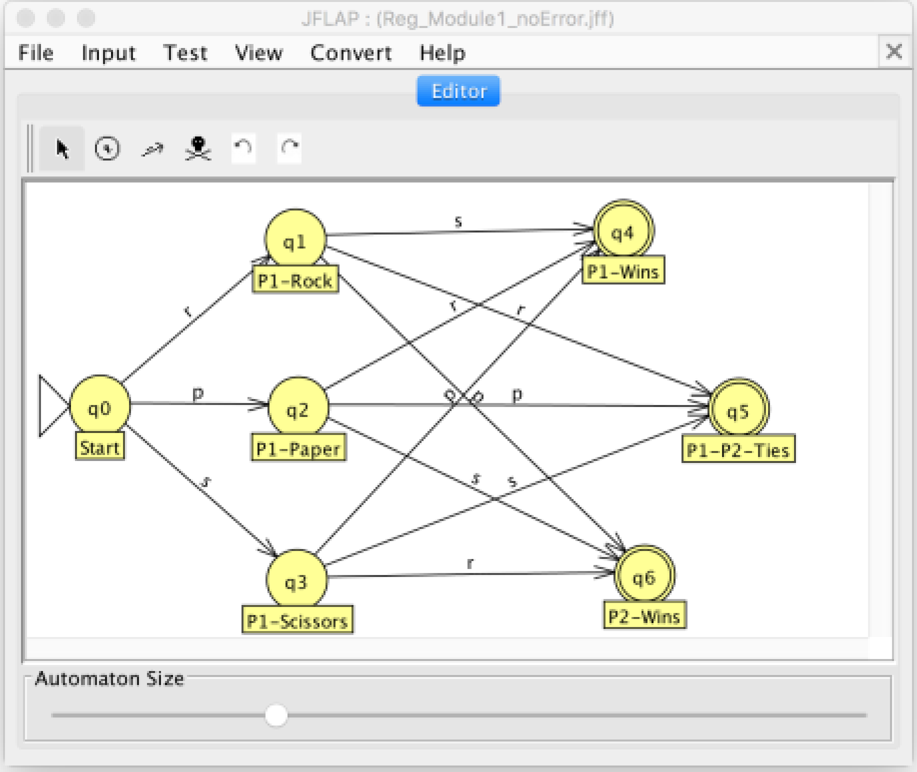


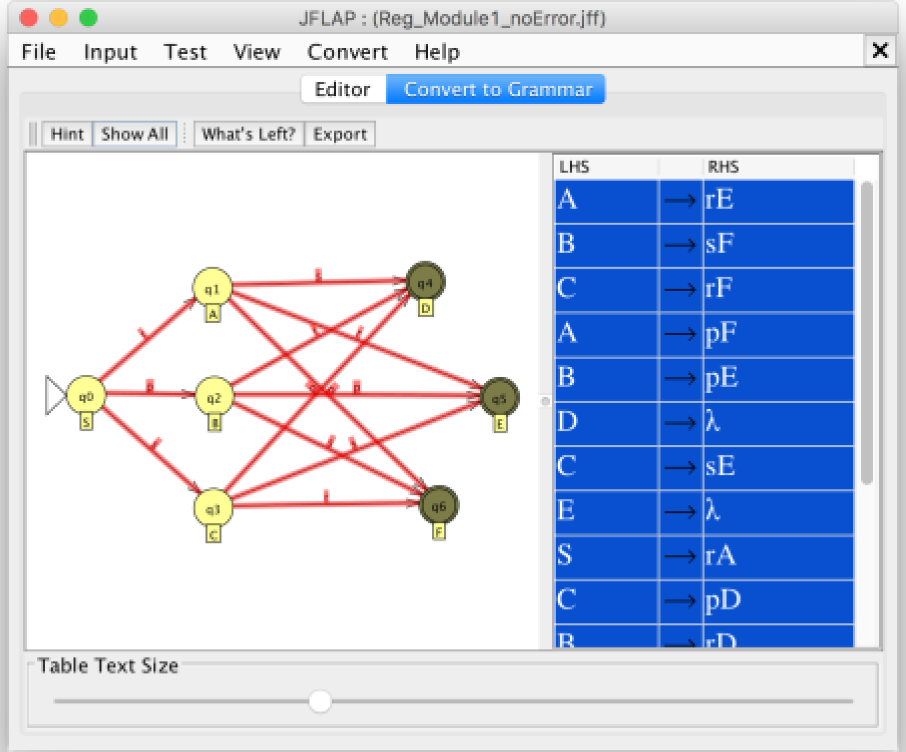
Input	Result
PP	Accept
SP	Accept
RS	Accept
RSP	Reject
RSSP	Reject
RT	Reject



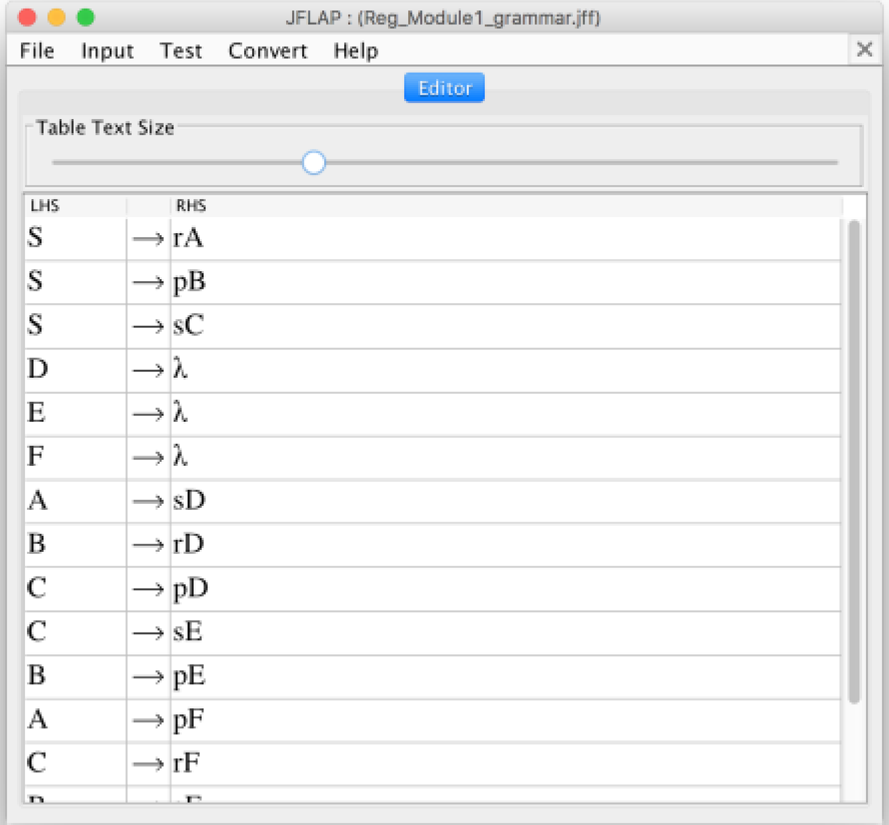


Capital letters are reserved for grammar variables.



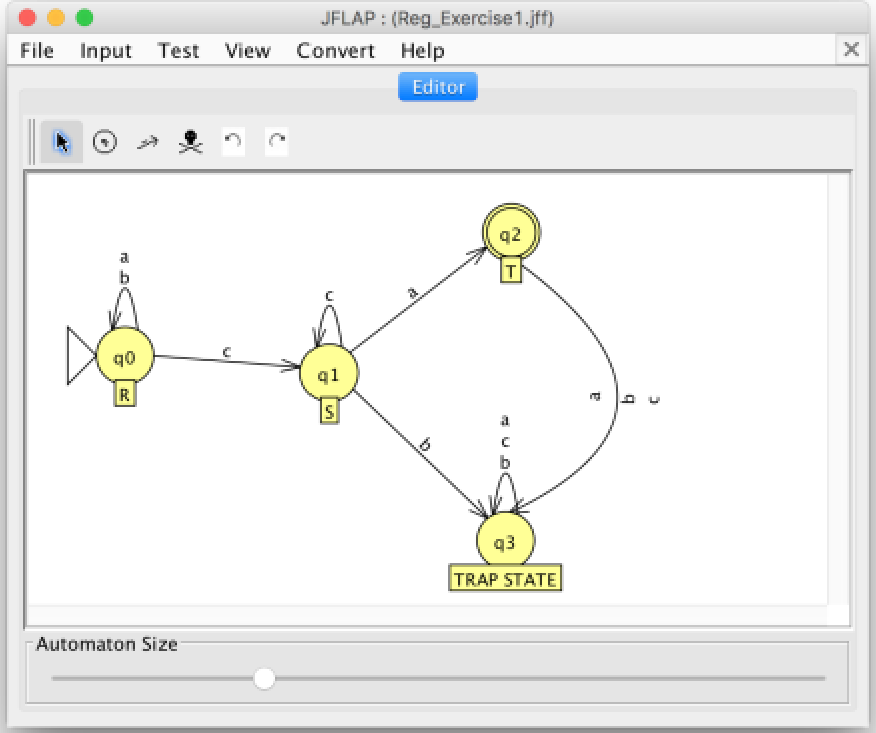


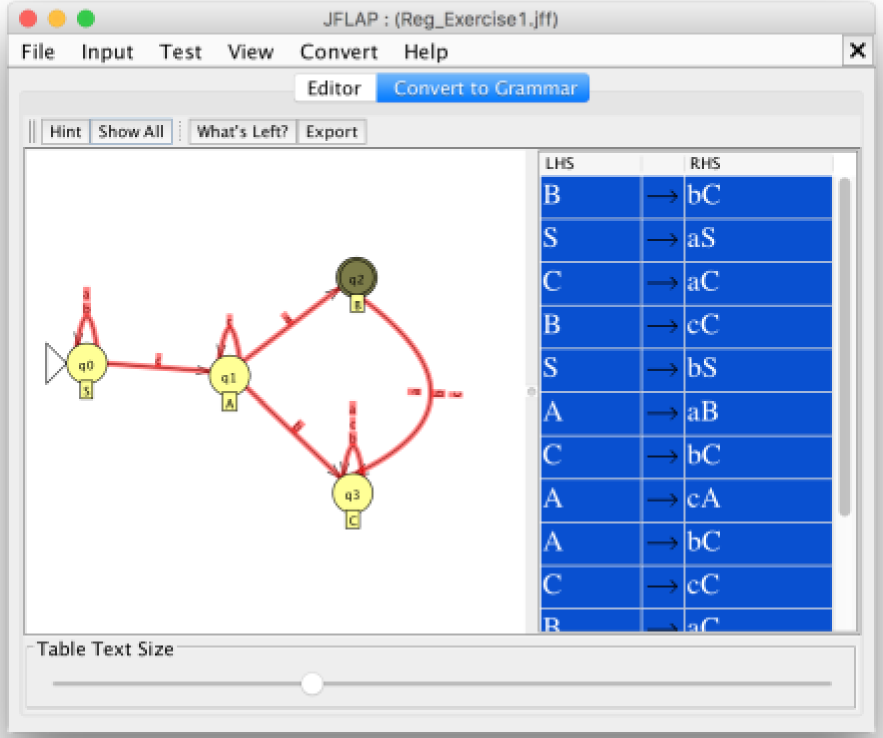
LHS		RHS
A	→	rE
B	→	sF
C	→	rF
A	→	pF
B	→	pE
D	→	λ
C	→	sE
E	→	λ
S	→	rA
C	→	pD
B	→	rD


JFLAP : (Reg_Module1_grammar.jff)
File Input Test Convert Help
Editor
Table Text Size

LHS		RHS
S	→	rA
S	→	pB
S	→	sC
D	→	λ
E	→	λ
F	→	λ
A	→	sD
B	→	rD
C	→	pD
C	→	sE
B	→	pE
A	→	pF
C	→	rF

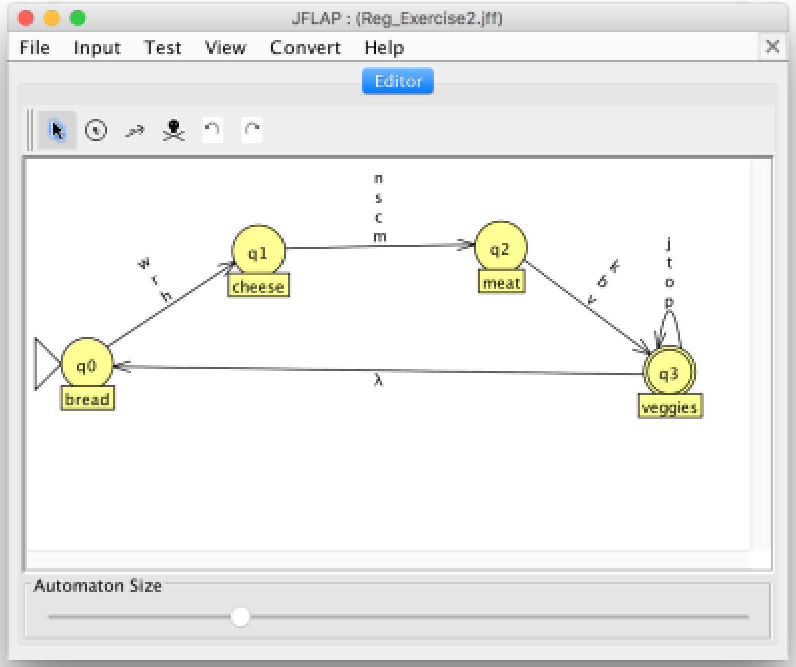
Exercise 1





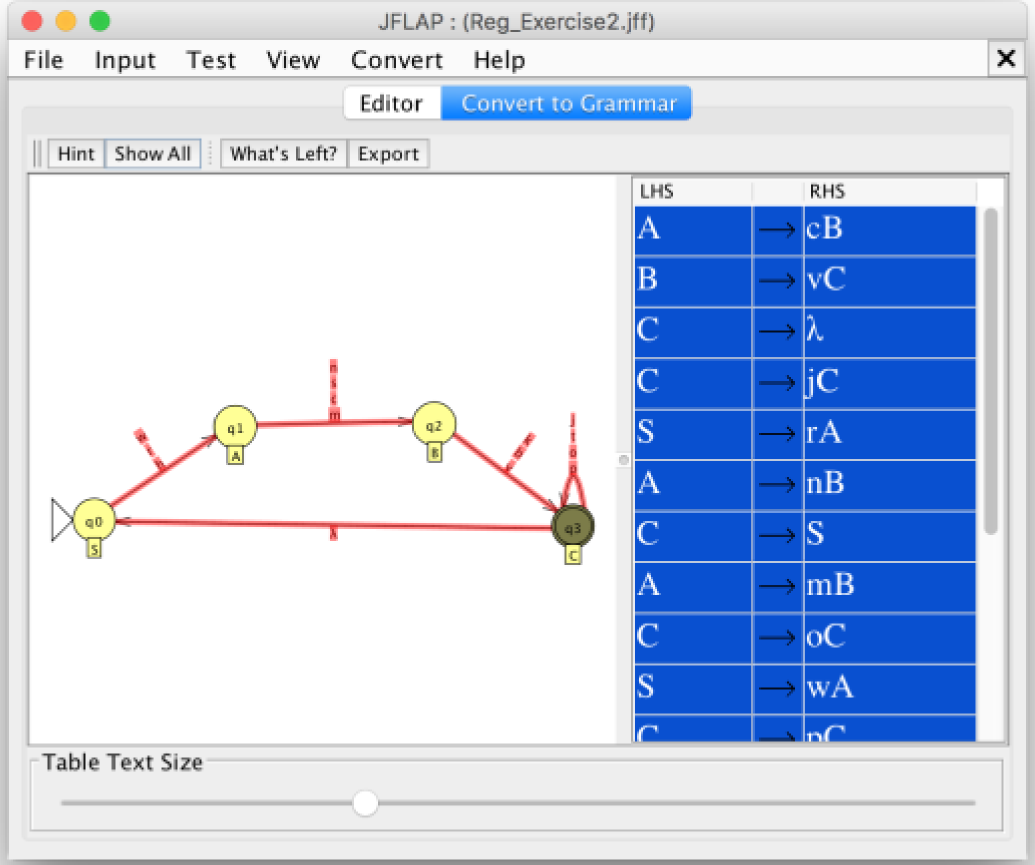
LHS		RHS
B	→	bC
S	→	aS
C	→	aC
B	→	cC
S	→	bS
A	→	aB
C	→	bC
A	→	cA
A	→	bC
C	→	cC
B	→	aC

Exercise 2



Questions to Think About

1. What is the length of the shortest string accepted by the DFA?
   3
2. How many such shortest strings are there? Why? (Hint: Think back to your discrete mathematics course.)
   3*4*3 = 36
3. What is the order in which components of a sandwich must be chosen? Why?
   Bread – cheese – meat – veggies, weil keine anderen Übergänge existieren
4. Can a valid string end with a comma? Why or why not?
   Nein, ist nicht im Alphabet.
5. How many variables will the corresponding regular grammar have? Why?
   14 – Anzahl Zeichen im Alphabet
6. How many production rules will the corresponding regular grammar have? Why?
   4 - Gruppen
7. How many lambda rules will the corresponding regular grammar have? Why. Remember that a **lambda rule** is a rule of the form A → λ, where A is a variable.
   1 – nächste Bestellung

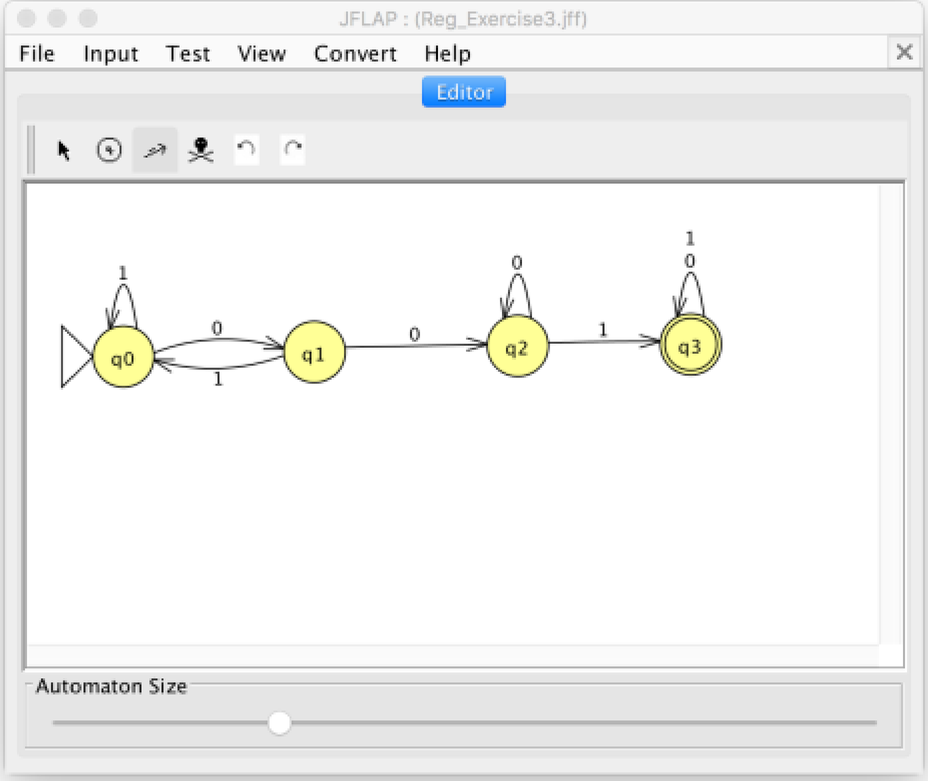


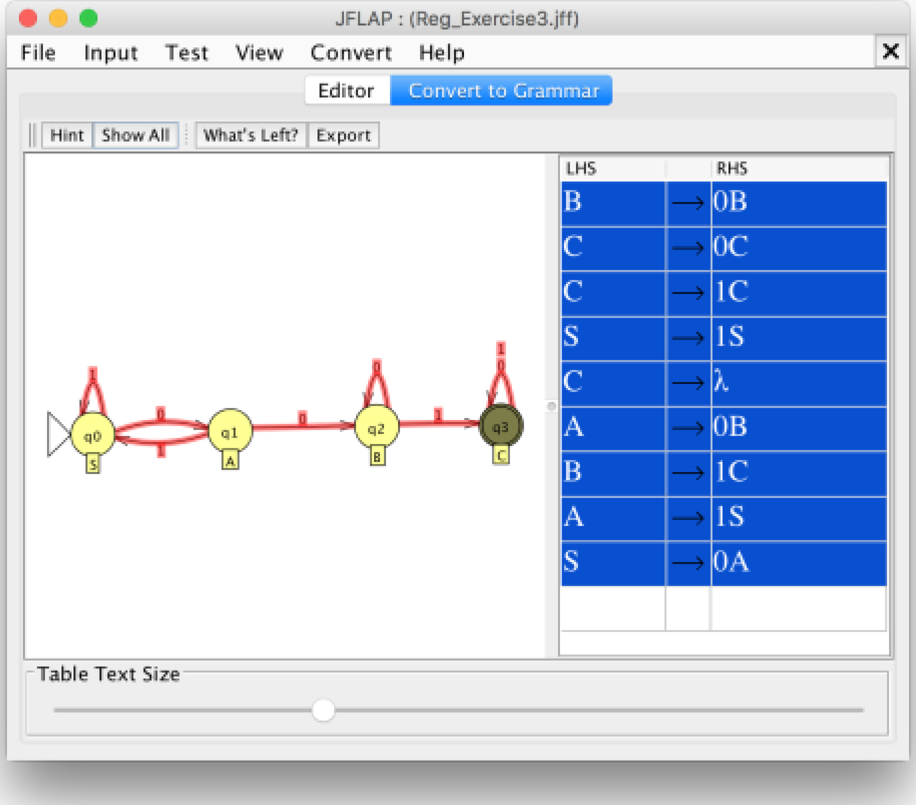
LHS		RHS
A	→	cB
B	→	vC
C	→	λ
C	→	jC
S	→	rA
A	→	nB
C	→	S
A	→	mB
C	→	oC
S	→	wA
C	→	nC

Questions to Think About

3. Does this grammar have recursive production rules? Why or why not?
   Ja – weil der letzte Schritt wieder den ersten ruft (iterativ?).
4. If it does have recursive production rules, how is the recursion stopped?
   Wenn nach den „veggies“ nichts mehr kommt.
5. What can you say about the number of strings in a language with a grammar with no recursive production rules?
   Die Anzahl Wörter ist begrenzt.

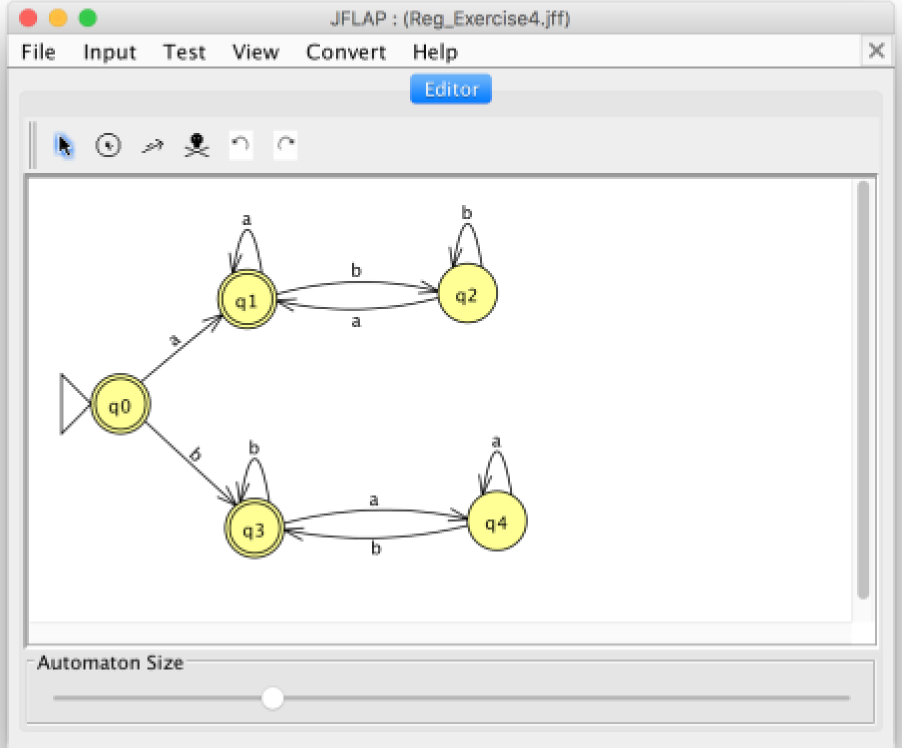
Exercise 3

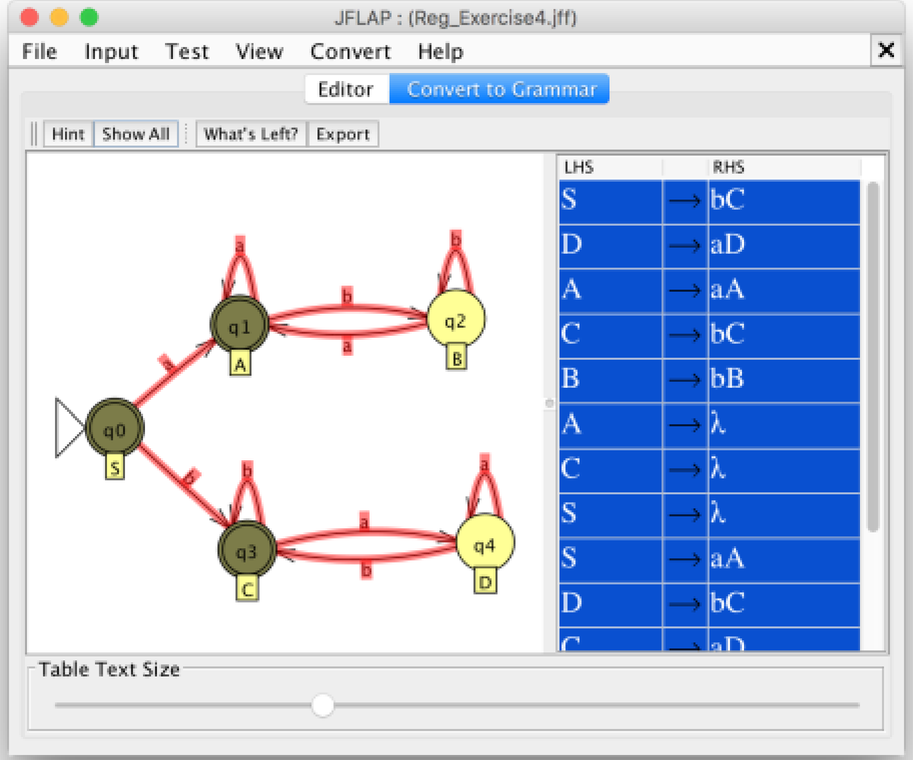

JFLAP : (Reg_Exercise3.jff)
File Input Test View Convert Help
Editor
q0
q1
q2
q3
0
1
0
1
1
0
0
1
Automaton Size



LHS		RHS
B	→	0B
C	→	0C
C	→	1C
S	→	1S
C	→	λ
A	→	0B
B	→	1C
A	→	1S
S	→	0A

Exercise 4





LHS		RHS
S	→	bC
D	→	aD
A	→	aA
C	→	bC
B	→	bB
A	→	λ
C	→	λ
S	→	λ
S	→	aA
D	→	bC
C	→	aD