

Práctica 05: Diseño y simulación de autómatas finitos en JFLAP

Computabilidad y Algoritmia
Grado en Ingeniería Informática
Universidad de La Laguna

Alejandro Rodríguez Rojas
alu0101317038@ull.edu.es

1. Diseño de DFAs	2
1.1. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de "a's" par.	2
1.2. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con longitud impar.	4
1.3. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de "a's" par o longitud impar.	6
1.4. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de "a's" par y longitud impar.	8
1.5. Diseñar un autómata finito determinista que reconozca cadenas w sobre el alfabeto $\{0, 1\}$ tales que $2 \leq w \leq 5$.	10
1.6. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ que tengan como mínimo dos ceros consecutivos.	12
1.7. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ que tengan como máximo dos ceros.	14
1.8. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ con longitud múltiplo de 3.	16
1.9. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ con longitud que no sea múltiplo de 3.	18
1.10. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{x, y, z\}$ que no contengan dos símbolos iguales consecutivos.	20
2. Diseño de NFAs	22
2.1. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por "a". A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	22
2.2. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que terminen en "bb". A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	26
2.3. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por "a" o terminen en "bb". A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	30
2.4. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por "a" y terminen en "bb". A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	34
2.5. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de "a's" par o longitud impar. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	38
2.6. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{x, y, z\}$ que contengan al menos dos símbolos iguales consecutivos. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	42
2.7. Diseñar un autómata finito no determinista que reconozca cadenas w sobre el alfabeto $\{x, y, z\}$, con $ w \geq 2$, tales que w empieza y termina por el mismo símbolo. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.	46
3. Modificación	50
3.1. Diseñar un NFA que reconozca las cadenas del lenguaje binario con dos ceros consecutivos y dos unos consecutivos.	50

1. Diseño de DFAs

1.1. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de “a’s” par.

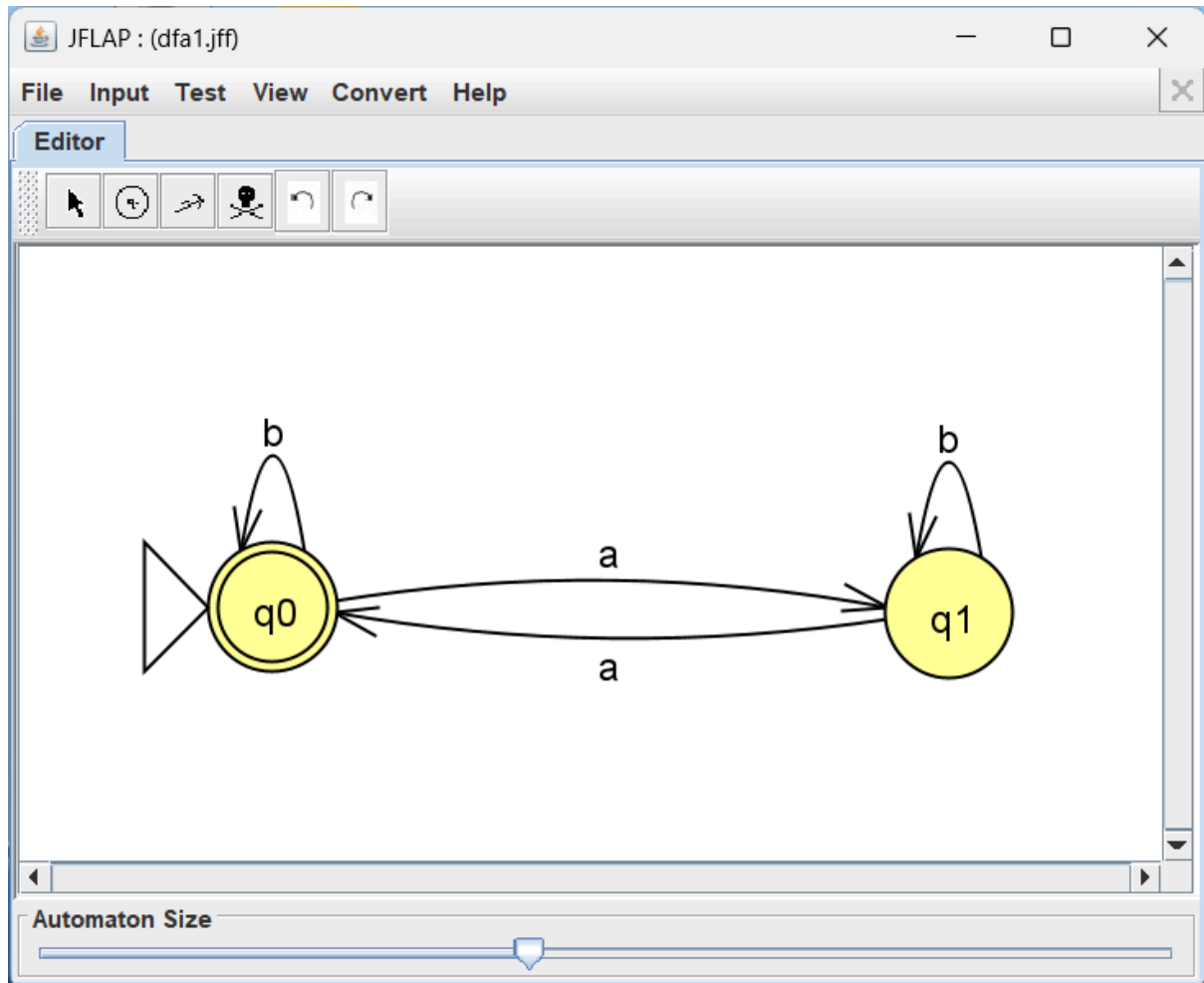


Figura 1: DFA que reconoce cadenas con número de “a’s” par sobre el alfabeto $\Sigma = \{a, b\}$

El estado q_0 representa el número par de “a’s” y la cantidad de “b’s” que sea, mientras que el estado q_1 representa una cantidad impar de “a’s” y la cantidad de “b’s” que sea

JFLAP : (dfa1.jff)

File Input Test View Convert Help

Editor Multiple Run

```

graph LR
    start(( )) --> q0((q0))
    q0 -- b --> q0
    q0 -- a --> q1((q1))
    q1 -- a --> q0
    q1 -- b --> q1
    q0 --> end((( )))
  
```

Table Text Size

Input	Result
a	Reject
aa	Accept
b	Accept
abba	Accept
ababaa	Accept
bba	Reject
aba	Accept
bbabb	Reject
aabaaa	Reject
bbababaaa	Reject
	Accept

Load Inputs Run Inputs Clear Enter Lamb

Figura 2: Cadenas de prueba para el DFA anterior

1.2. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con longitud impar.

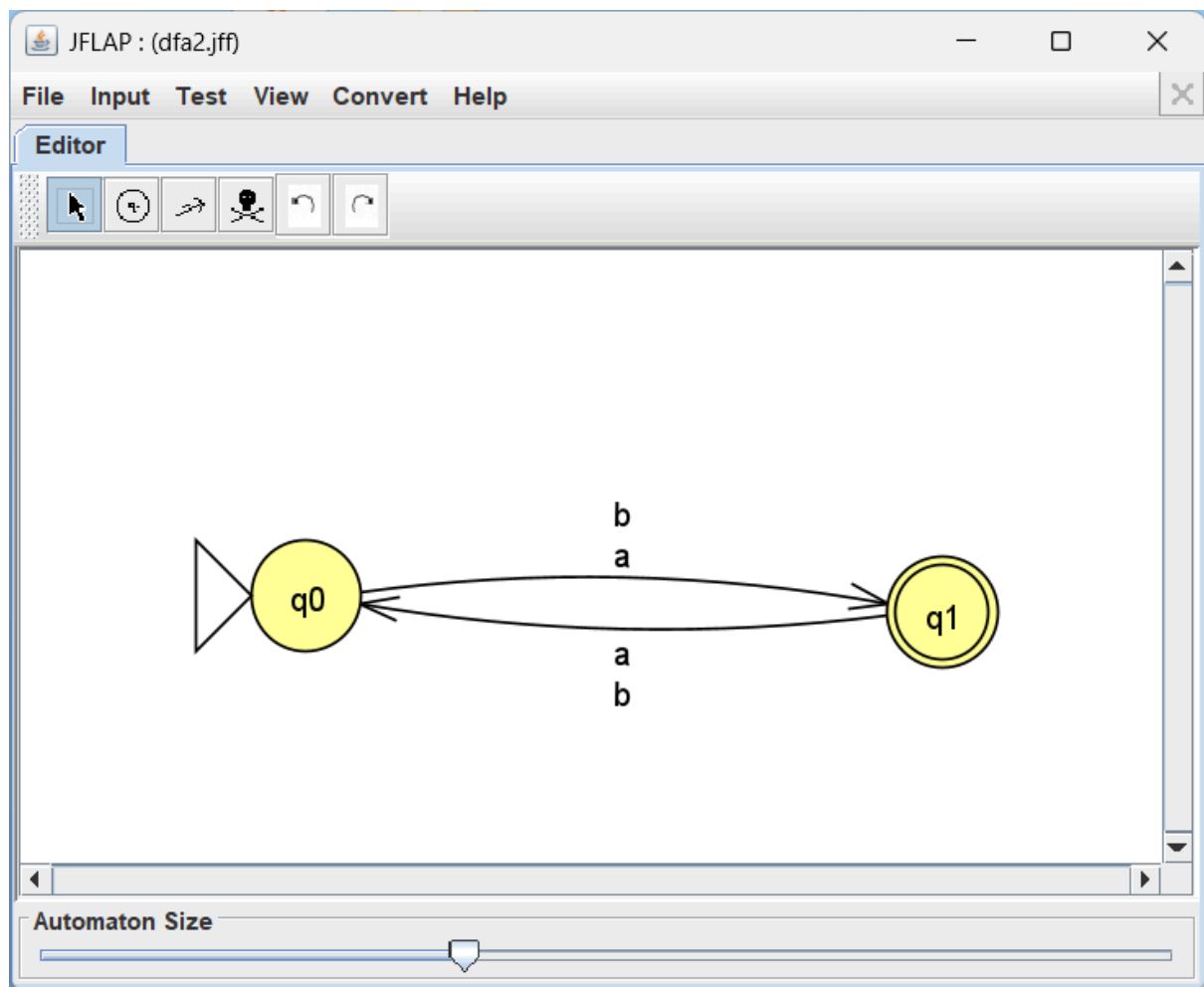


Figura 3: DFA que reconoce cadenas de longitud impar sobre el alfabeto $\Sigma = \{a, b\}$

JFLAP : (dfa2.jff)

File Input Test View Convert Help

Editor Multiple Run

```

graph LR
    start(( )) --> q0((q0))
    q0 -- a --> q1(((q1)))
    q1 -- b --> q0
    q0 -- b --> q1
    q1 -- a --> q0
  
```

Table Text Size

Input	Result
a	Accept
aa	Reject
b	Accept
abba	Reject
ababaa	Reject
bba	Accept
aba	Accept
bbabb	Accept
aabaaa	Reject
bbababaaa	Accept
	Reject

Load Inputs Run Inputs Clear Enter Lamb

Figura 4: Cadenas de prueba para el DFA anterior

1.3. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de “a’s” par o longitud impar.

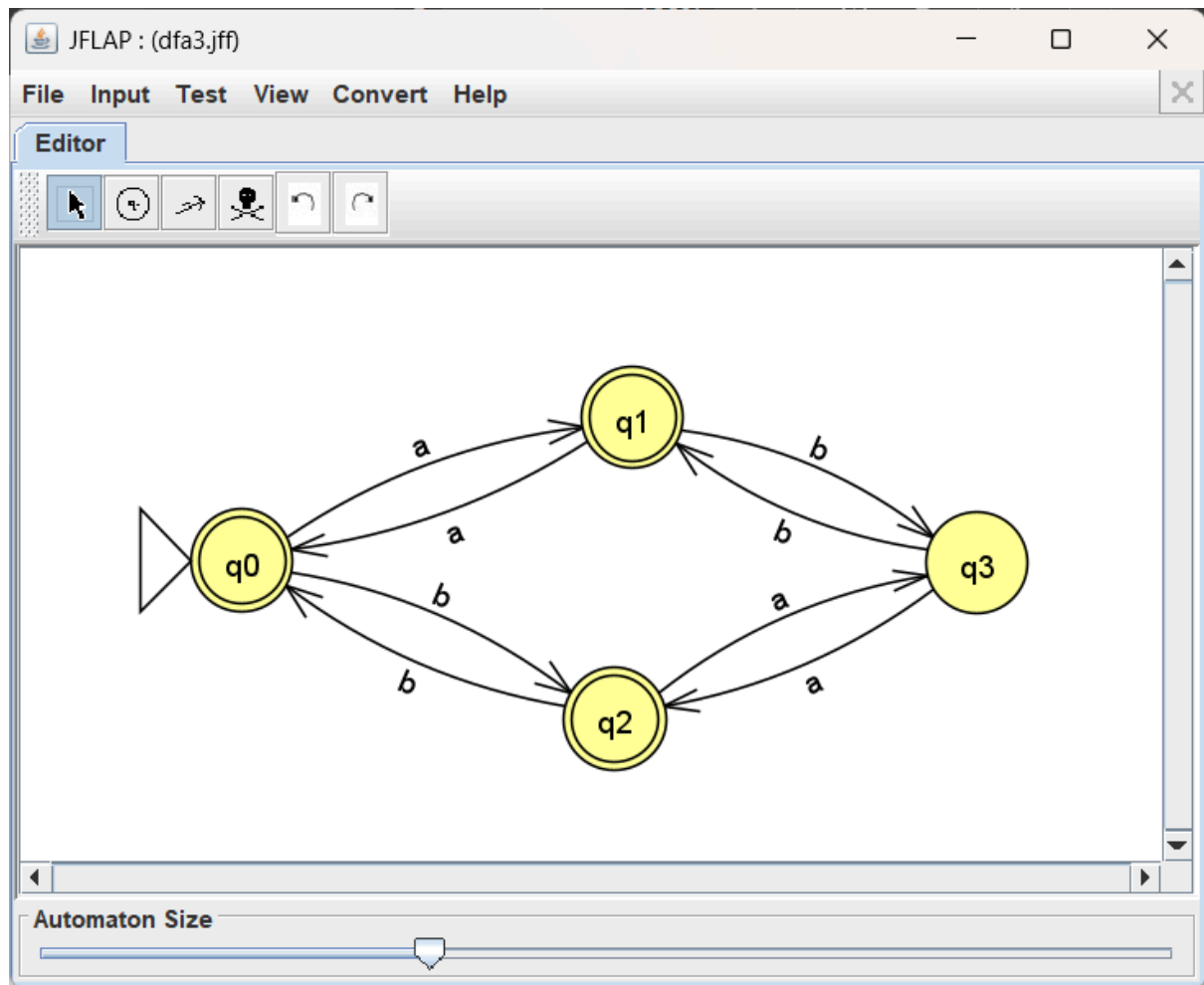


Figura 5: DFA que reconoce cadenas con número de “a’s” par o longitud impar sobre el alfabeto $\Sigma = \{a, b\}$

JFLAP : (dfa3.jff)

File Input Test View Convert Help

Editor Multiple Run

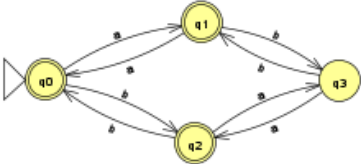


Table Text Size

Input	Result
aba	Accept
b	Accept
aabb	Accept
a	Accept
aaabb	Accept
aab	Accept
abababa	Accept
bbab	Reject
aaab	Reject
bbbbab	Reject
baabab	Reject
bbbaaaa	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

Figura 6: Cadenas de prueba para el DFA anterior

1.4. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de “a’s” par y longitud impar.

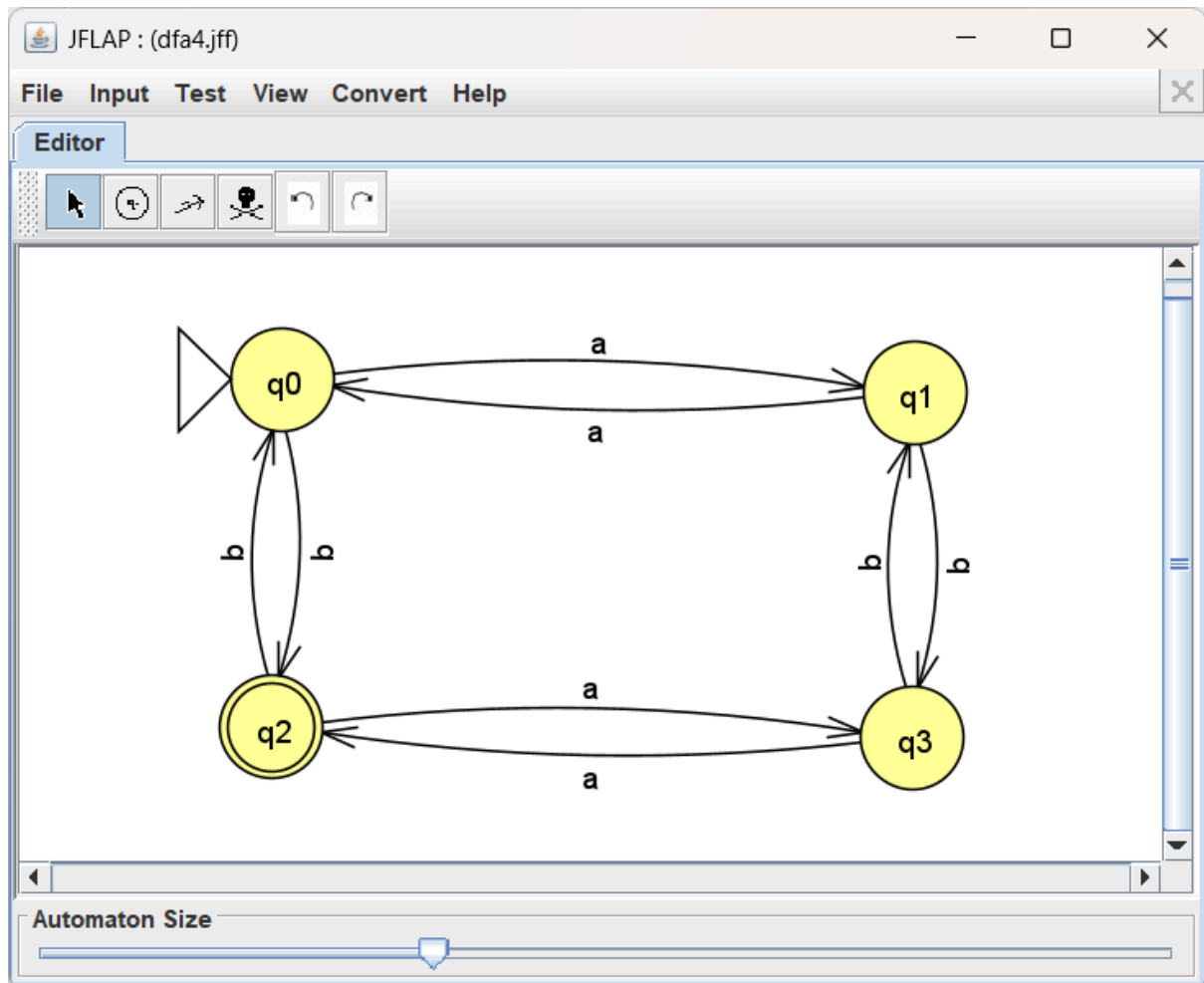


Figura 7: DFA que reconoce cadenas con número de “a’s” par y longitud impar sobre el alfabeto $\Sigma = \{a, b\}$

JFLAP : (dfa4.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
aba	Accept
b	Accept
aabb	Reject
a	Reject
aaabb	Reject
aab	Accept
abababa	Accept
bbab	Reject
aaab	Reject
bbbbab	Reject
baababa	Accept
bbbaaaaa	Accept
aaaabbb	Accept
abbbbbaaaba	Accept
bb	Reject

Load Inputs Run Inputs Clear Enter Lamb

Figura 8: Cadenas de prueba para el DFA anterior

1.5. Diseñar un autómata finito determinista que reconozca cadenas w sobre el alfabeto $\{0, 1\}$ tales que $2 \leq |w| \leq 5$.

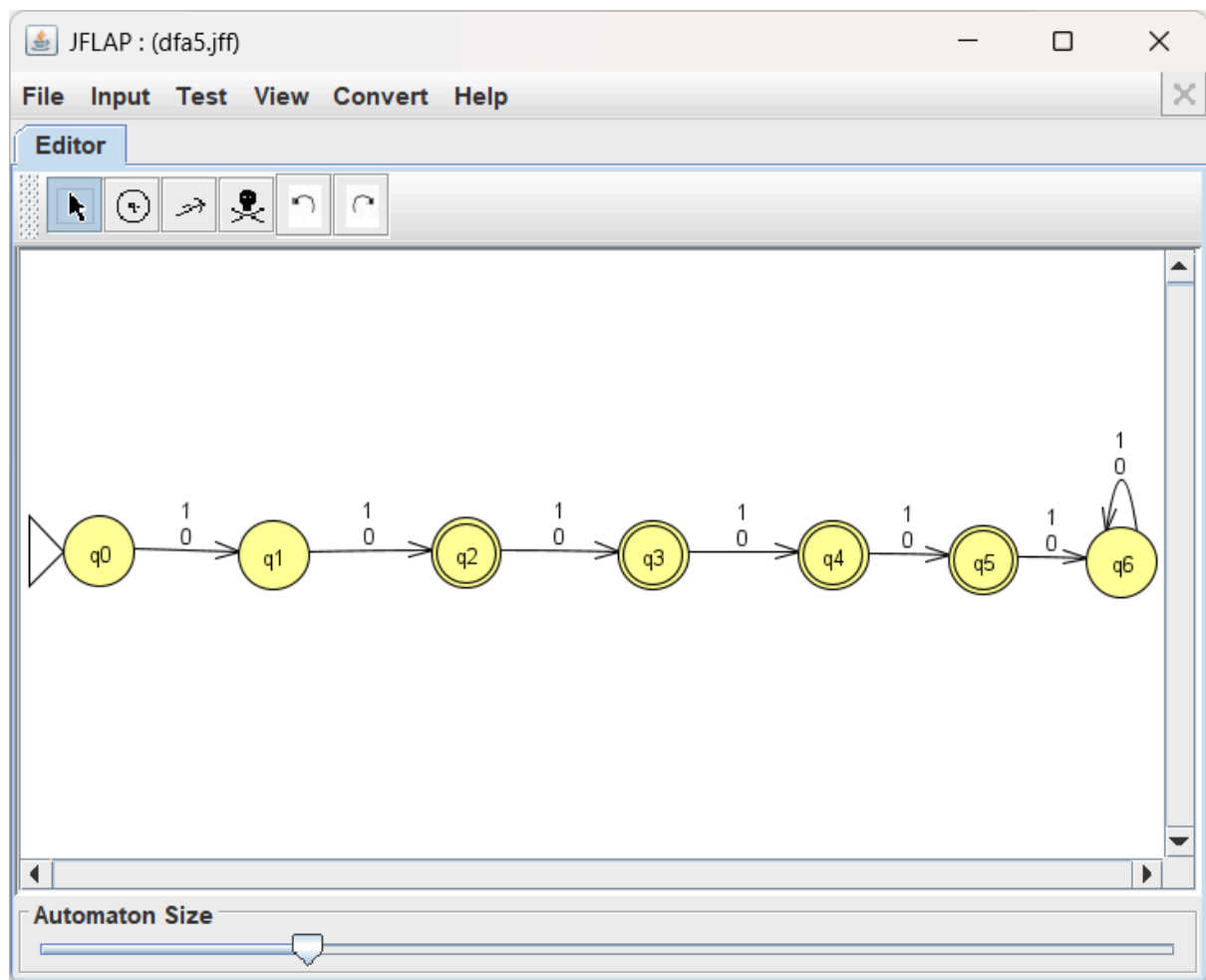


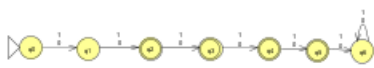
Figura 9: DFA que reconoce cadenas w sobre el alfabeto $\{0, 1\}$ tales que $2 \leq |w| \leq 5$.

JFLAP : (dfa5.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size



Input	Result
0	Reject
11	Accept
101	Accept
0011	Accept
01111	Accept
10100	Accept
1100	Accept
1101	Accept
010101	Reject
1100111	Reject
11010111	Reject
101001000	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

Figura 10: Cadenas de prueba para el DFA anterior.

1.6. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ que tengan como mínimo dos ceros consecutivos.

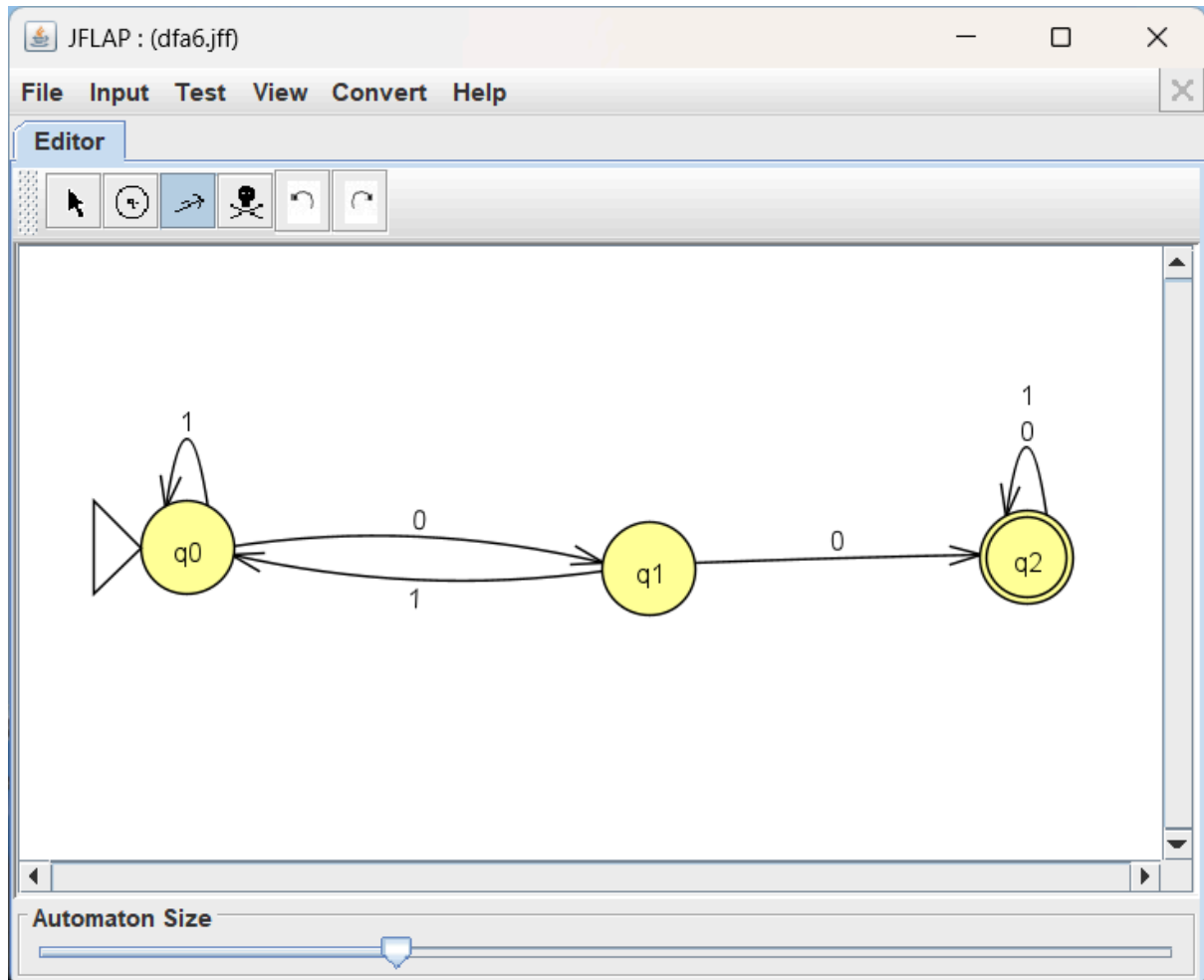


Figura 11: DFA que reconoce cadenas que tengan como mínimo dos ceros consecutivos sobre el alfabeto $\{0, 1\}$.

JFLAP : (dfa6.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
01	Reject
101001	Accept
0111	Reject
101000	Accept
11011	Reject
111010101001	Accept
01110	Reject
10100	Accept
11011	Reject
1001	Accept

Load Inputs Run Inputs Clear Enter Lamb

Figura 12: Cadenas de prueba para el DFA anterior.

1.7. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ que tengan como máximo dos ceros.

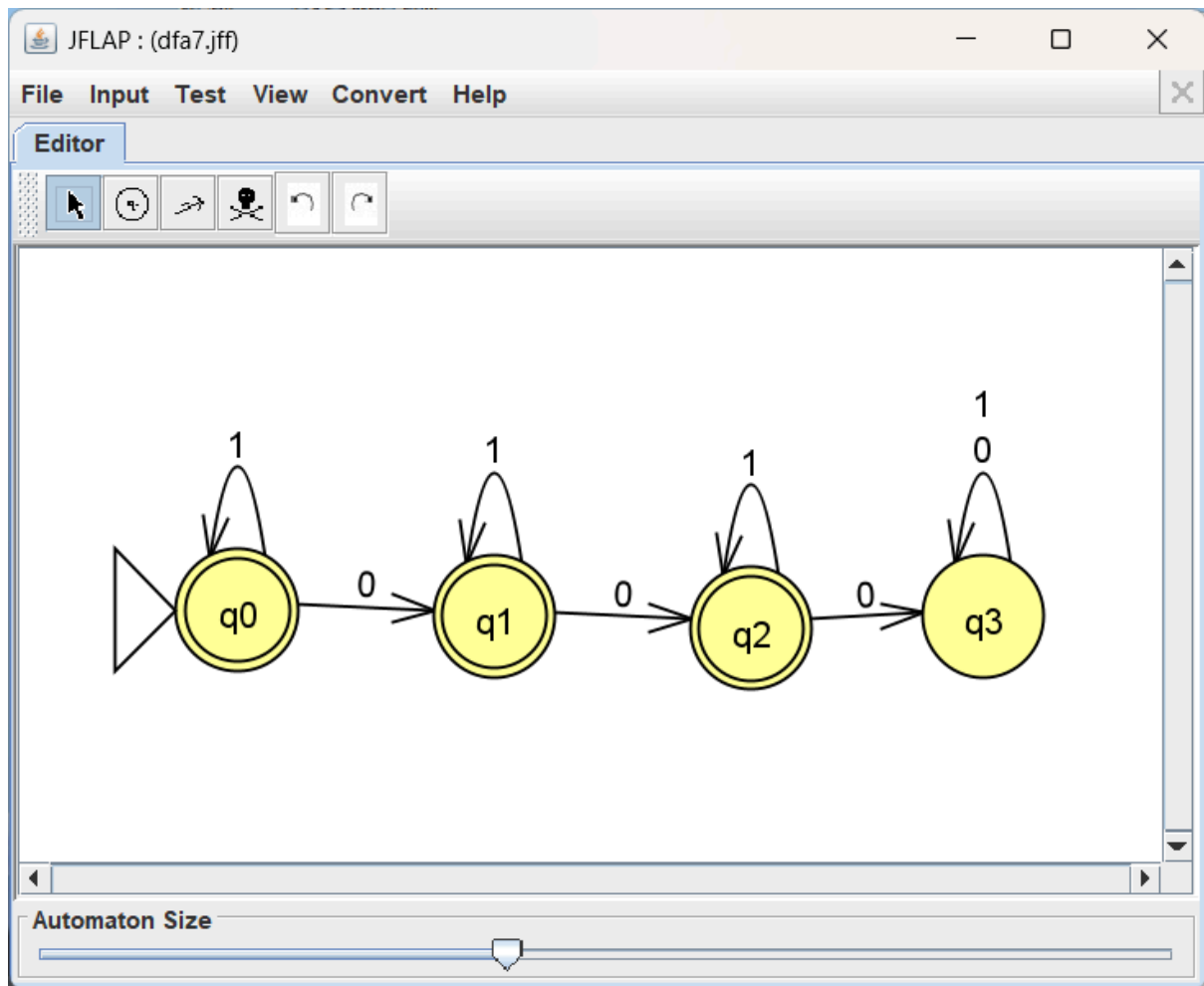


Figura 13: DFA que reconoce cadenas que tengan como máximo dos ceros sobre el alfabeto $\{0, 1\}$

JFLAP : (dfa7.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
01	Accept
101001	Reject
0111	Accept
101000	Reject
11011	Accept
111010101001	Reject
01110	Accept
10100	Reject
11011	Accept
1001	Accept

Load Inputs Run Inputs Clear Enter Lambda View Trace

Figura 14: Cadenas de prueba para el DFA anterior.

1.8. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ con longitud múltiplo de 3.

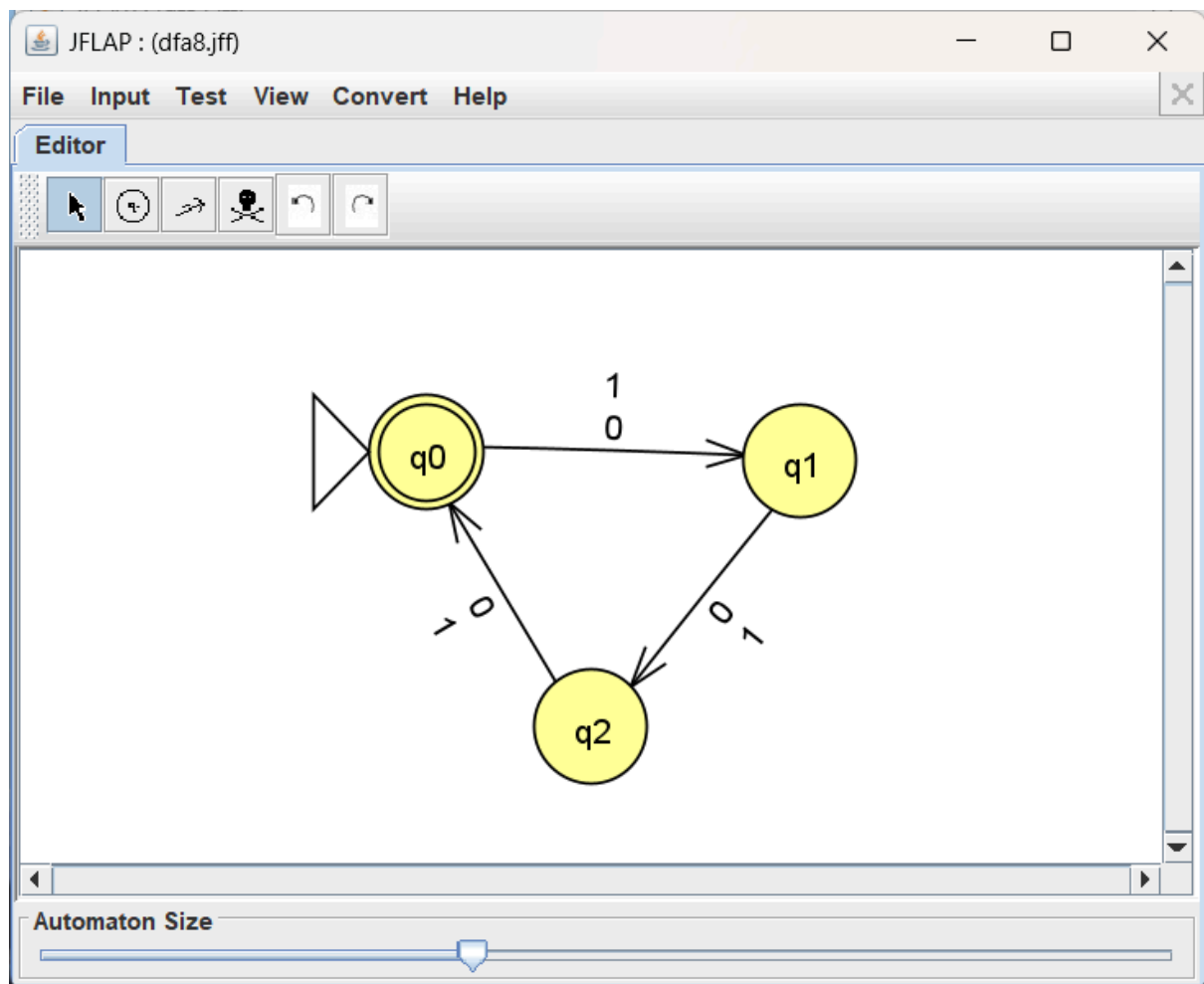


Figura 15: DFA que reconoce cadenas con longitud múltiplo de 3 sobre el alfabeto $\{0, 1\}$

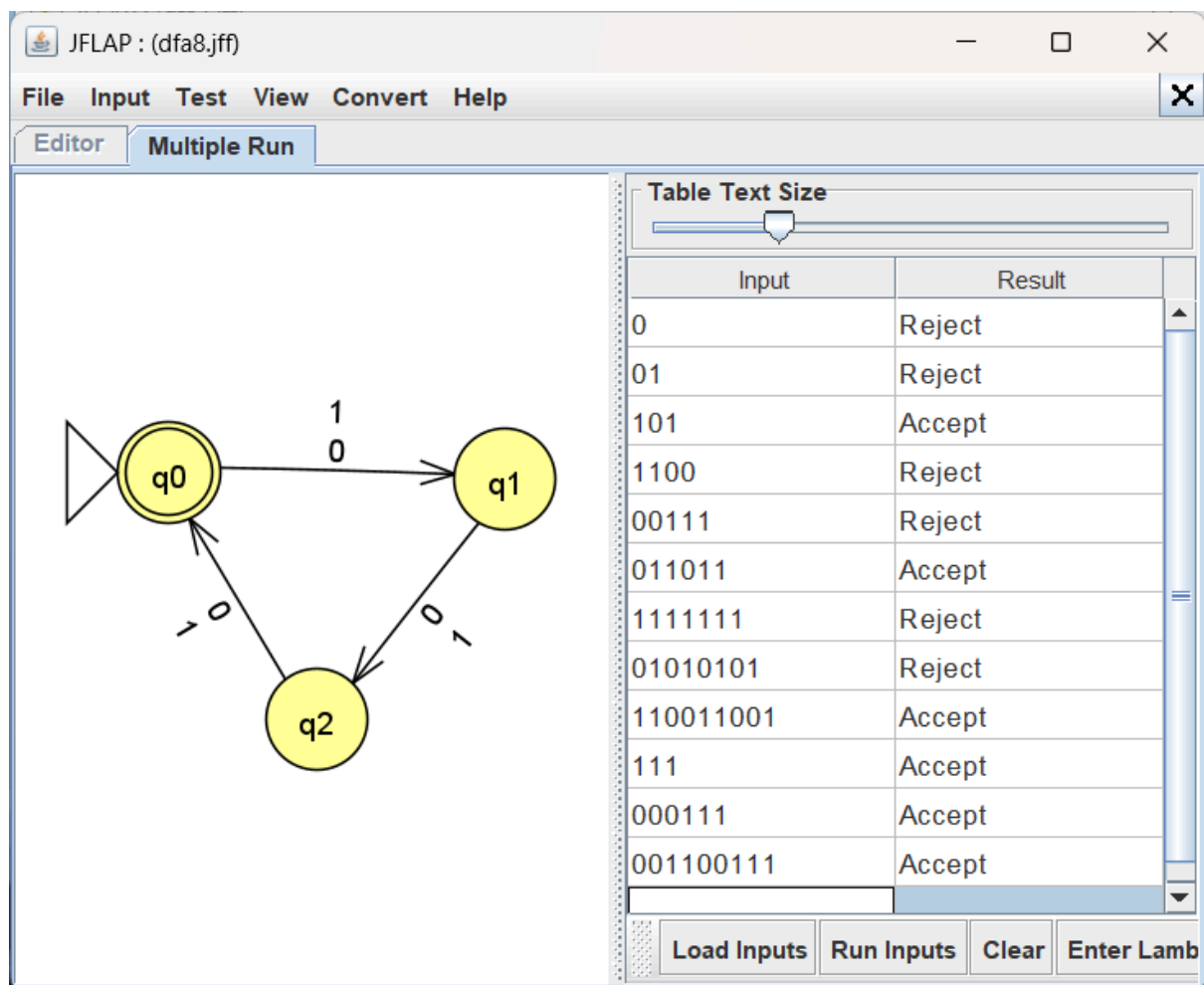


Figura 16: Cadenas de prueba para el DFA anterior.

1.9. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{0, 1\}$ con longitud que no sea múltiplo de 3.

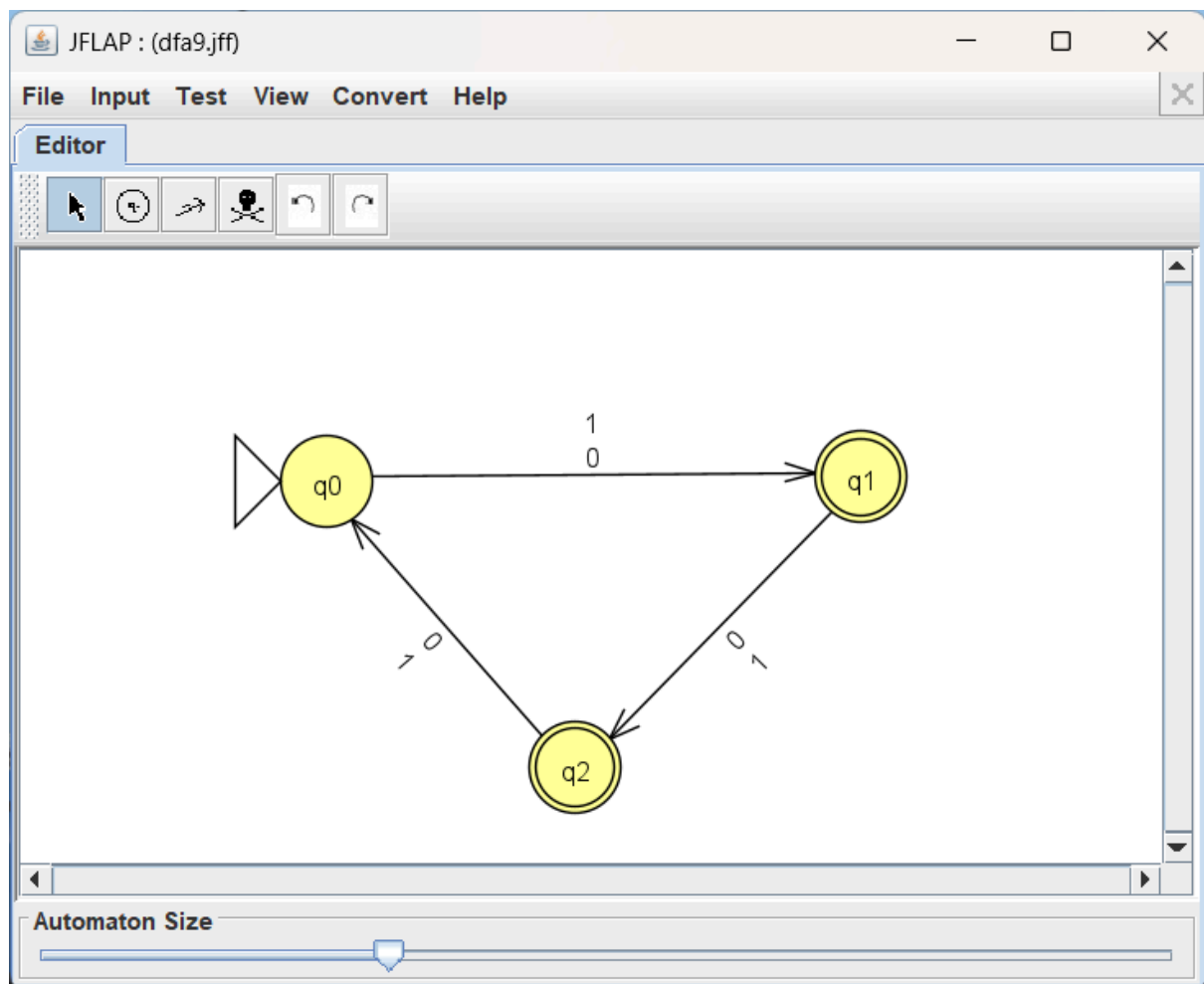


Figura 17: DFA que reconoce cadenas con longitud que no sea múltiplo de 3 sobre el alfabeto $\{0, 1\}$

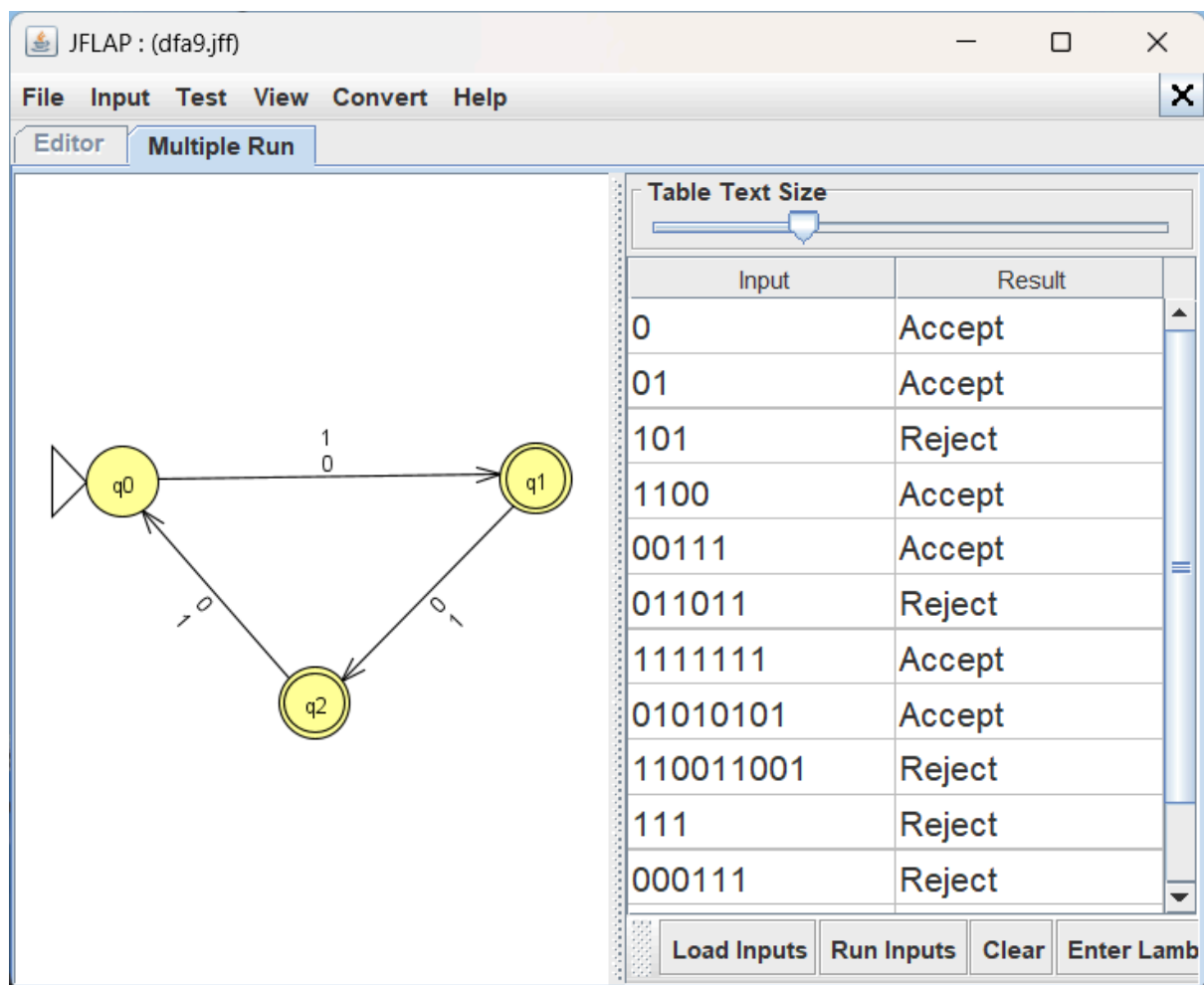


Figura 18: Cadenas de prueba para el DFA anterior.

1.10. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto $\{x, y, z\}$ que no contengan dos símbolos iguales consecutivos.

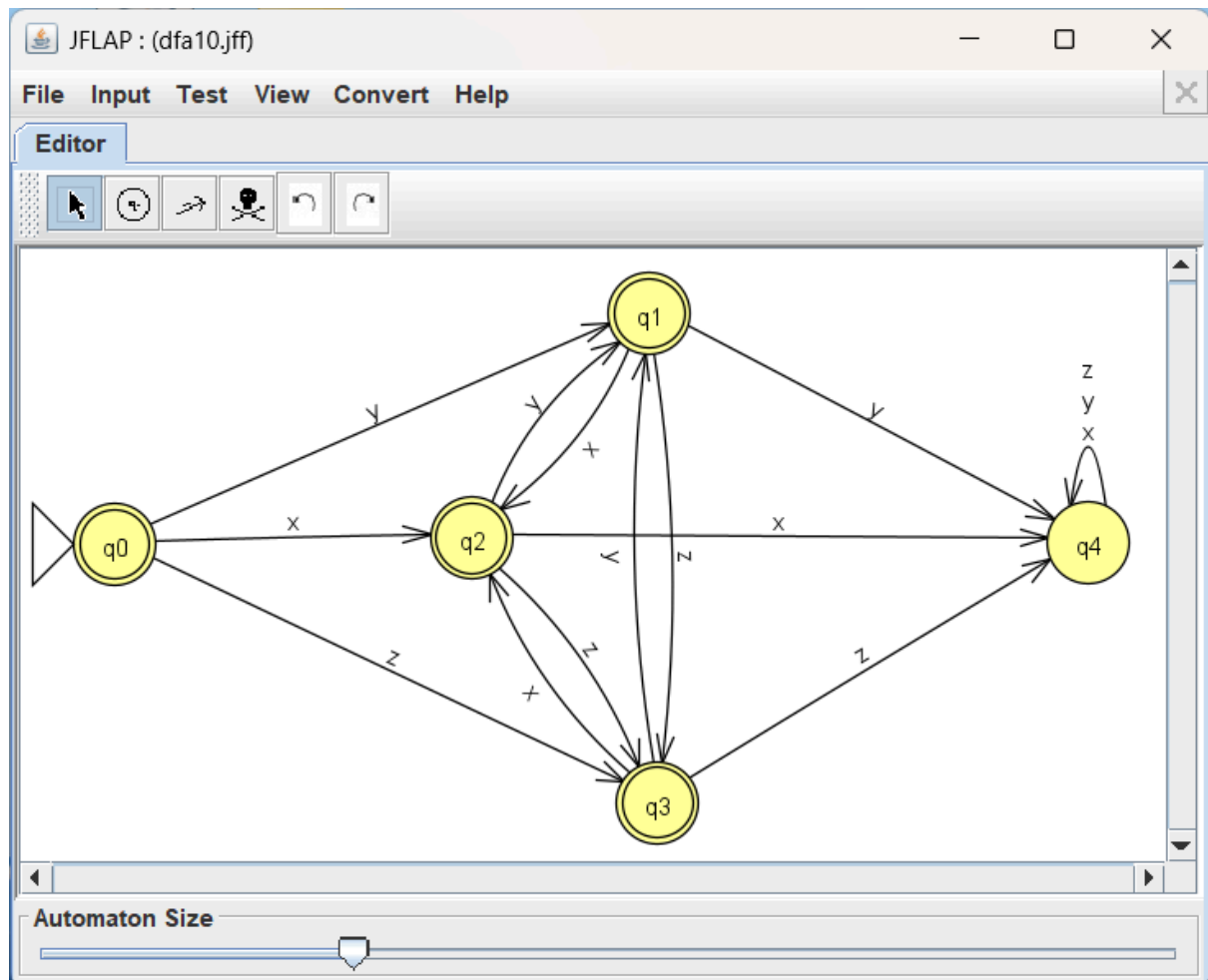


Figura 19: DFA que reconoce cadenas que no contengan dos símbolos iguales consecutivos sobre el alfabeto $\{x, y, z\}$

JFLAP : (dfa10.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
xyz	Accept
yzzx	Reject
zyxzxz	Accept
yzxzy	Accept
yyz	Reject
yzxxzy	Reject
yxzyxz	Accept
zzxxyy	Reject
zxyxyxz	Accept
zzxzyy	Reject
yz	Accept
xxx	Reject

Load Inputs Run Inputs Clear Enter Lambda View Trace

Figura 20: Cadenas de prueba para el DFA anterior.

2. Diseño de NFAs

2.1. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por "a". A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

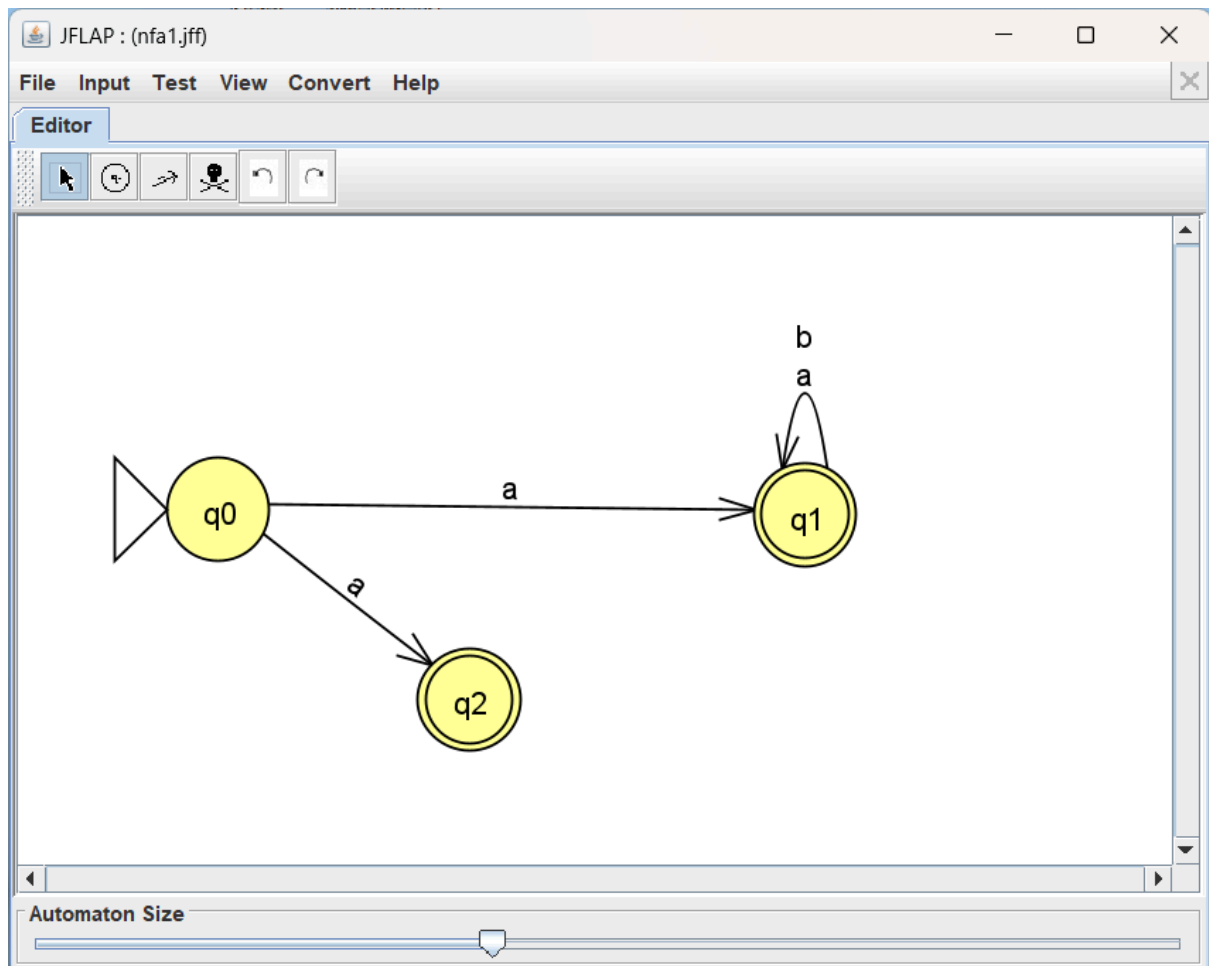


Figura 21: NFA que reconoce cadenas que empiecen por "a" sobre el alfabeto $\Sigma = \{a, b\}$

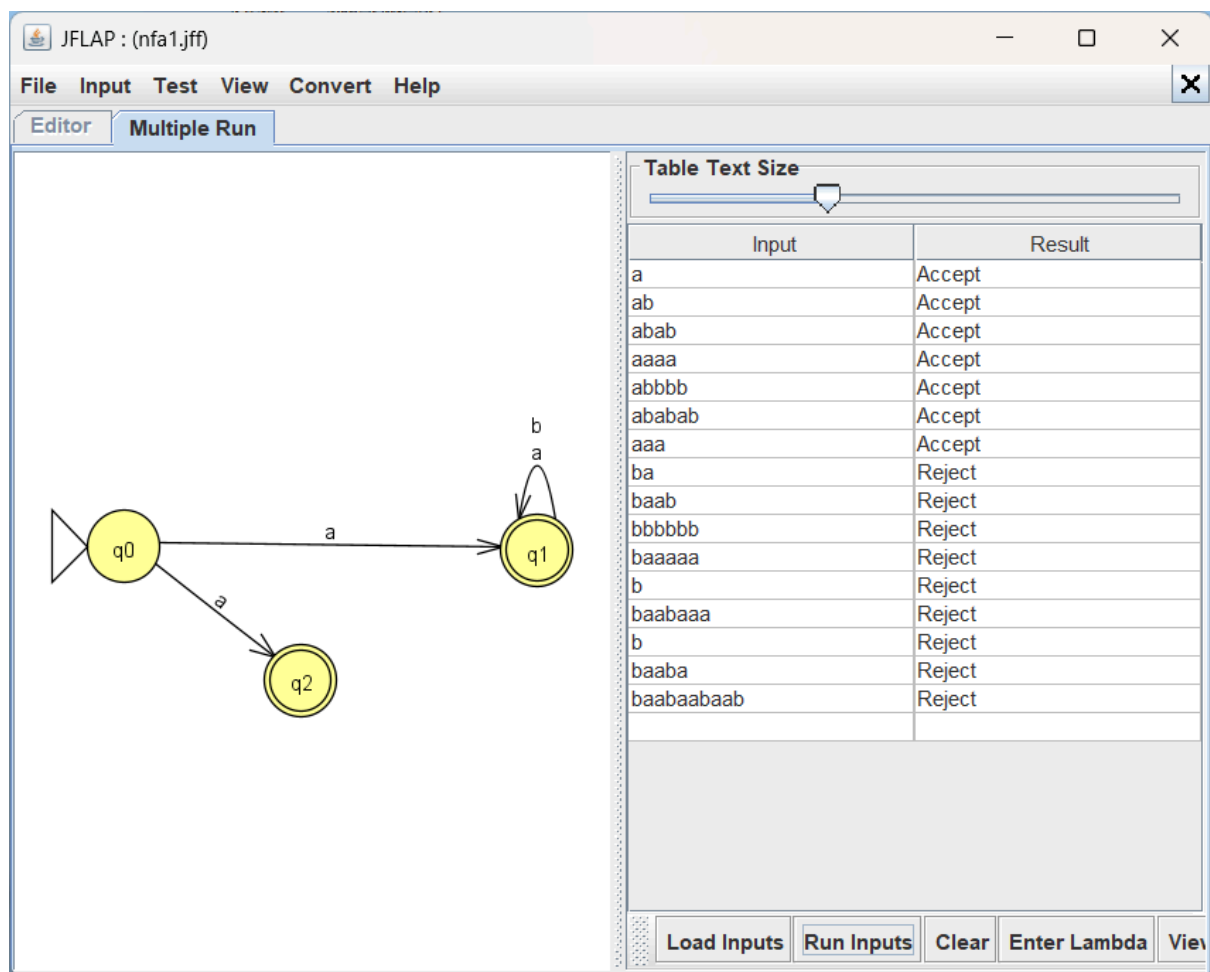


Figura 22: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos (sin ϵ -clausura ya que no hay epsilon transiciones) para pasar el NFA a un DFA equivalente:

$S = \{q_0\} = A$
 $\delta(A, a) = \{q_1, q_2\} = B$
 $\delta(A, b) = \emptyset$
 $\delta(B, a) = \{q_1\} = C$
 $\delta(B, b) = \{q_1\} = C$
 $\delta(C, a) = \{q_1\} = C$
 $\delta(C, b) = \{q_1\} = C$

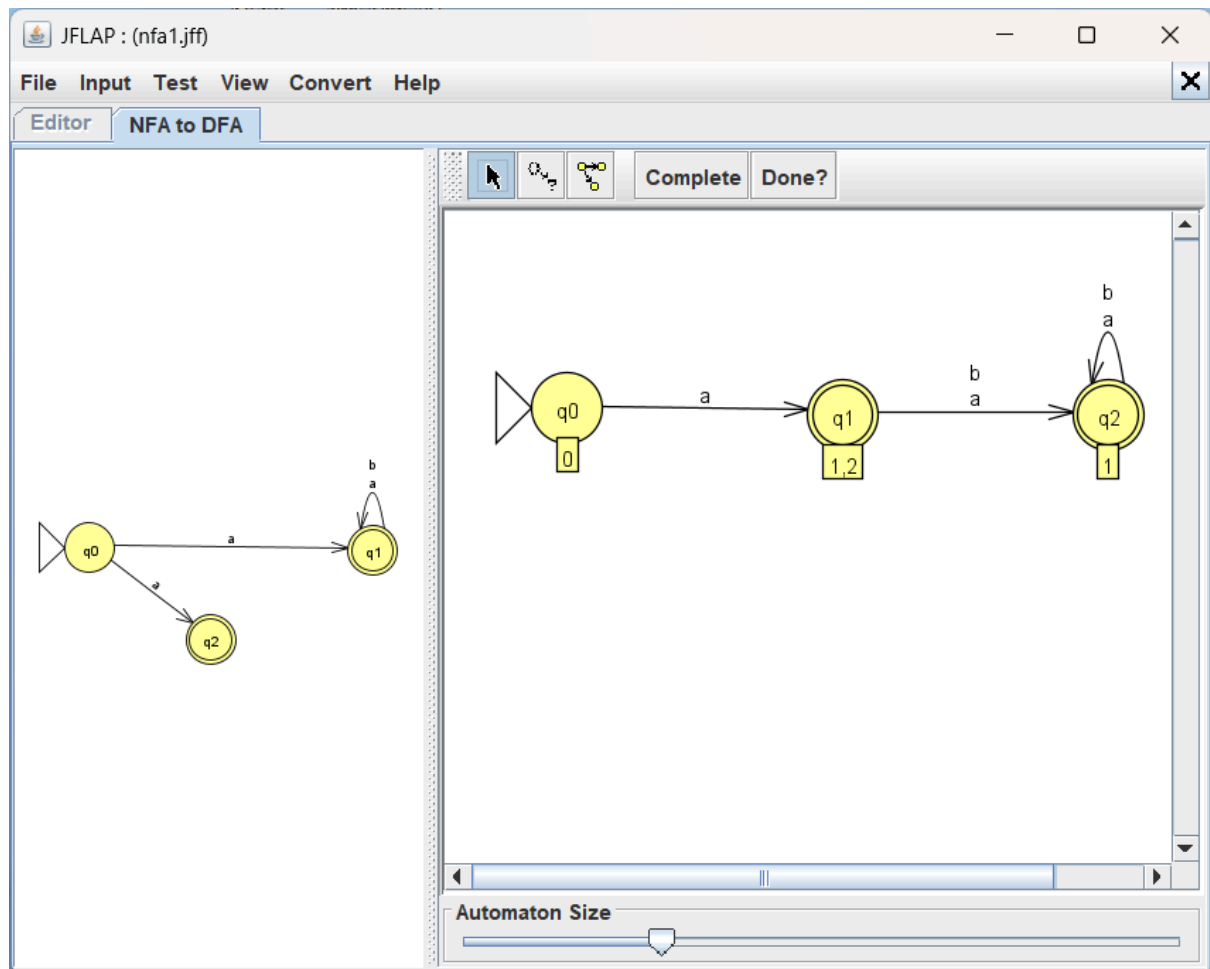


Figura 23: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$$\pi = \{\{B, C\}, \{A, \emptyset\}\}$$

$$\pi' = \{\{B, C\}, \{A\}, \{\emptyset\}\}$$

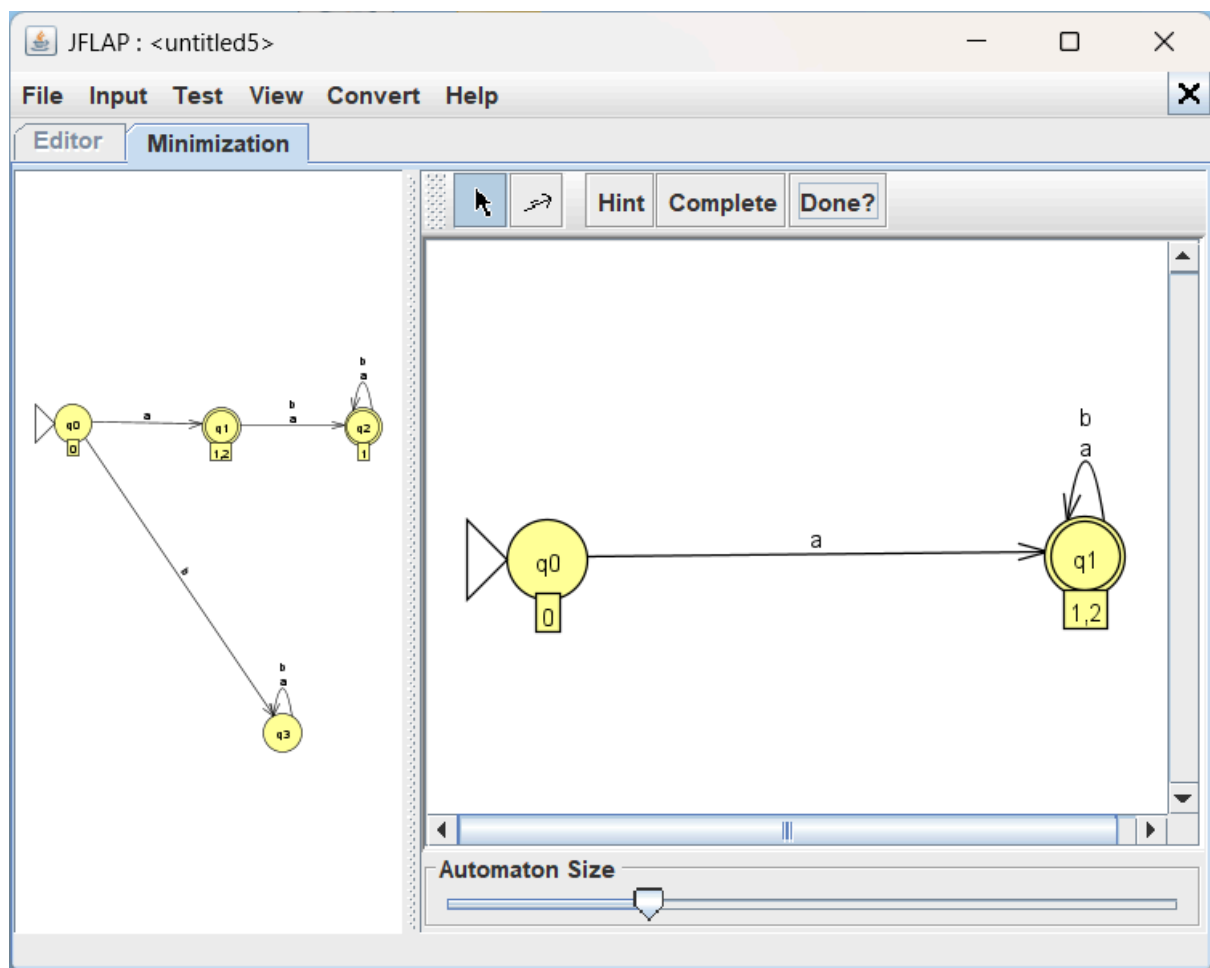


Figura 24: DFA mínimo equivalente.

2.2. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que terminen en “bb”. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

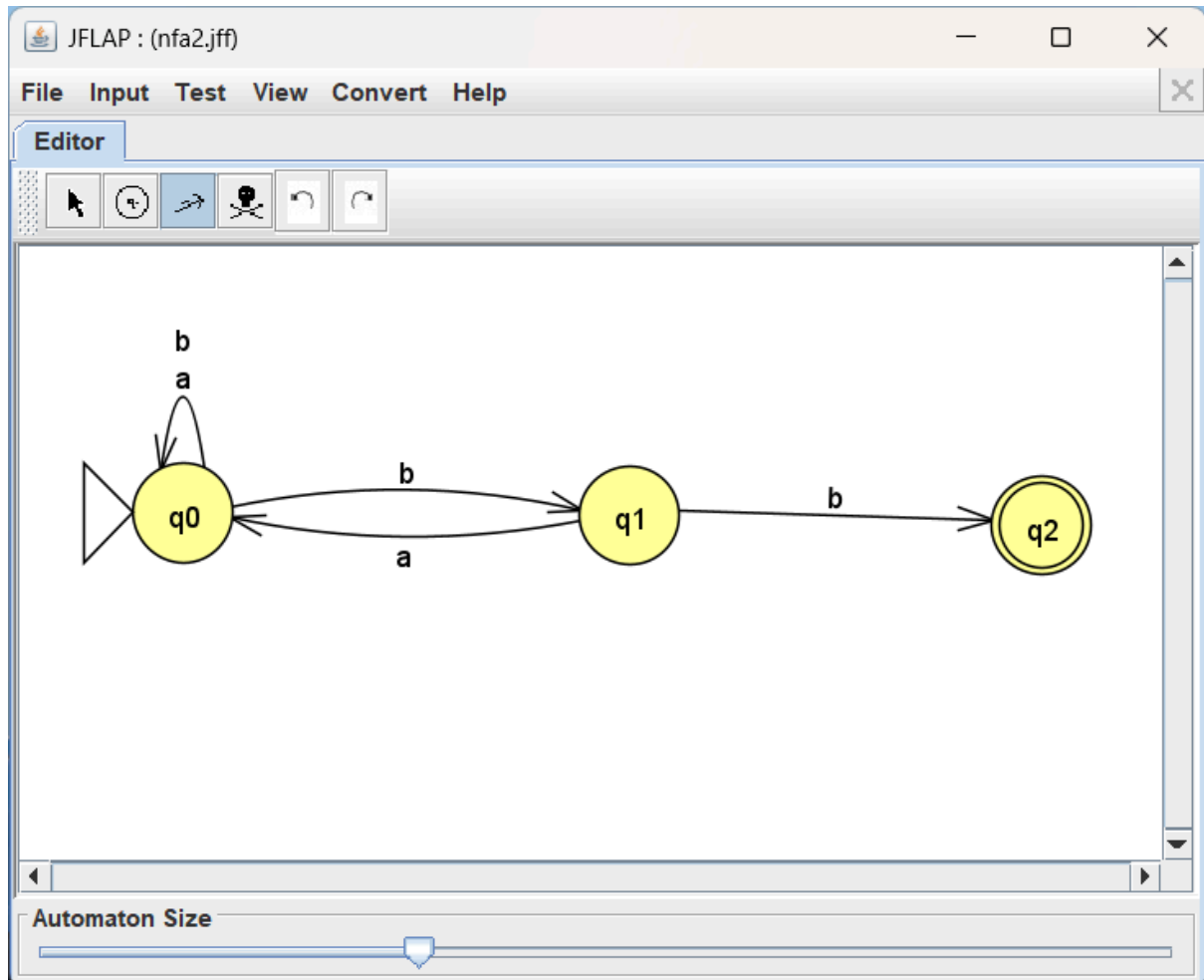


Figura 25: NFA que reconoce cadenas que terminen en “bb” sobre el alfabeto $\Sigma = \{a, b\}$

JFLAP : (nfa2.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
aba	Reject
bba	Reject
abb	Accept
bbabb	Accept
bababa	Reject
babaabb	Accept
abba	Reject
bb	Accept
babb	Accept
bbb	Accept
baabb	Accept
bababaa	Reject
aaa	Reject
bbbbabbbb	Accept
abbabb	Accept

Load Inputs Run Inputs Clear Enter Lamb

Figura 26: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos (sin ϵ -clausura ya que no hay epsilon transiciones) para pasar el NFA a un DFA equivalente:

$$S = \{q_0\} = A$$

$$\delta(A, a) = \{q_0\} = A$$

$$\delta(A, b) = \{q_0, q_1\} = B$$

$$\delta(B, a) = \{q_0\} = A$$

$$\delta(B, b) = \{q_0, q_1, q_2\} = C$$

$$\delta(C, a) = \{q_0\} = A$$

$$\delta(C, b) = \{q_0, q_1, q_2\} = C$$

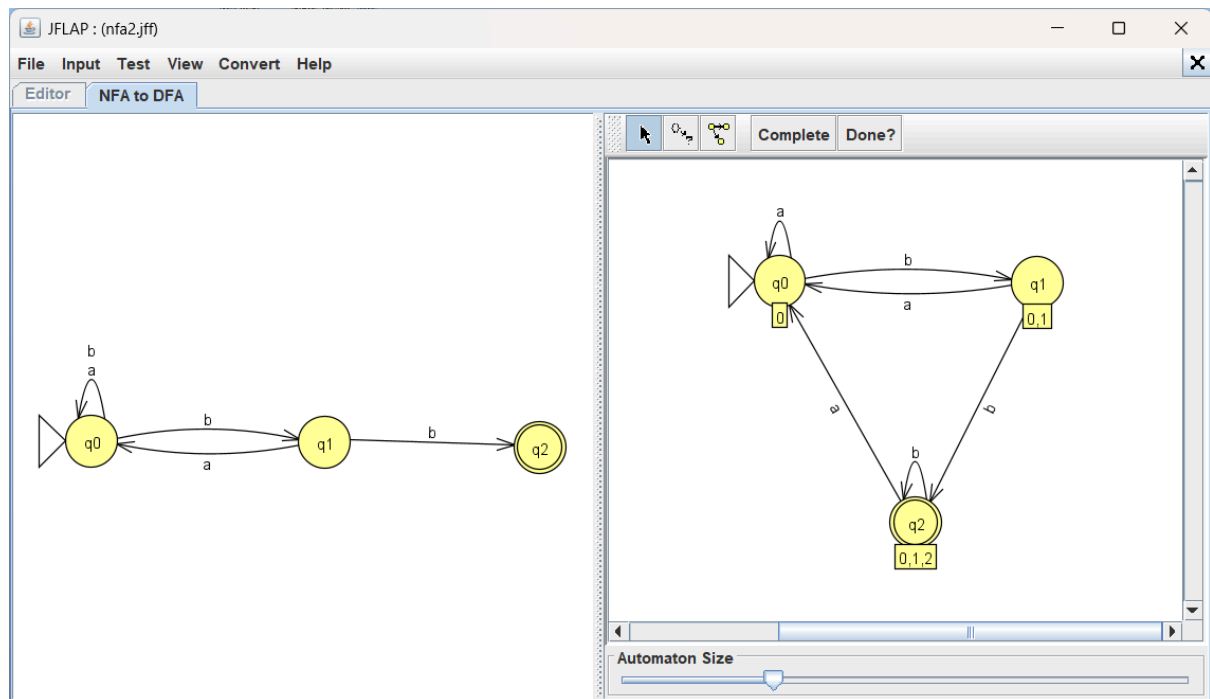


Figura 27: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$$\pi = \{\{C\}, \{A, B\}\}$$

$$\pi' = \{\{C\}, \{A\}\{B\}\}$$

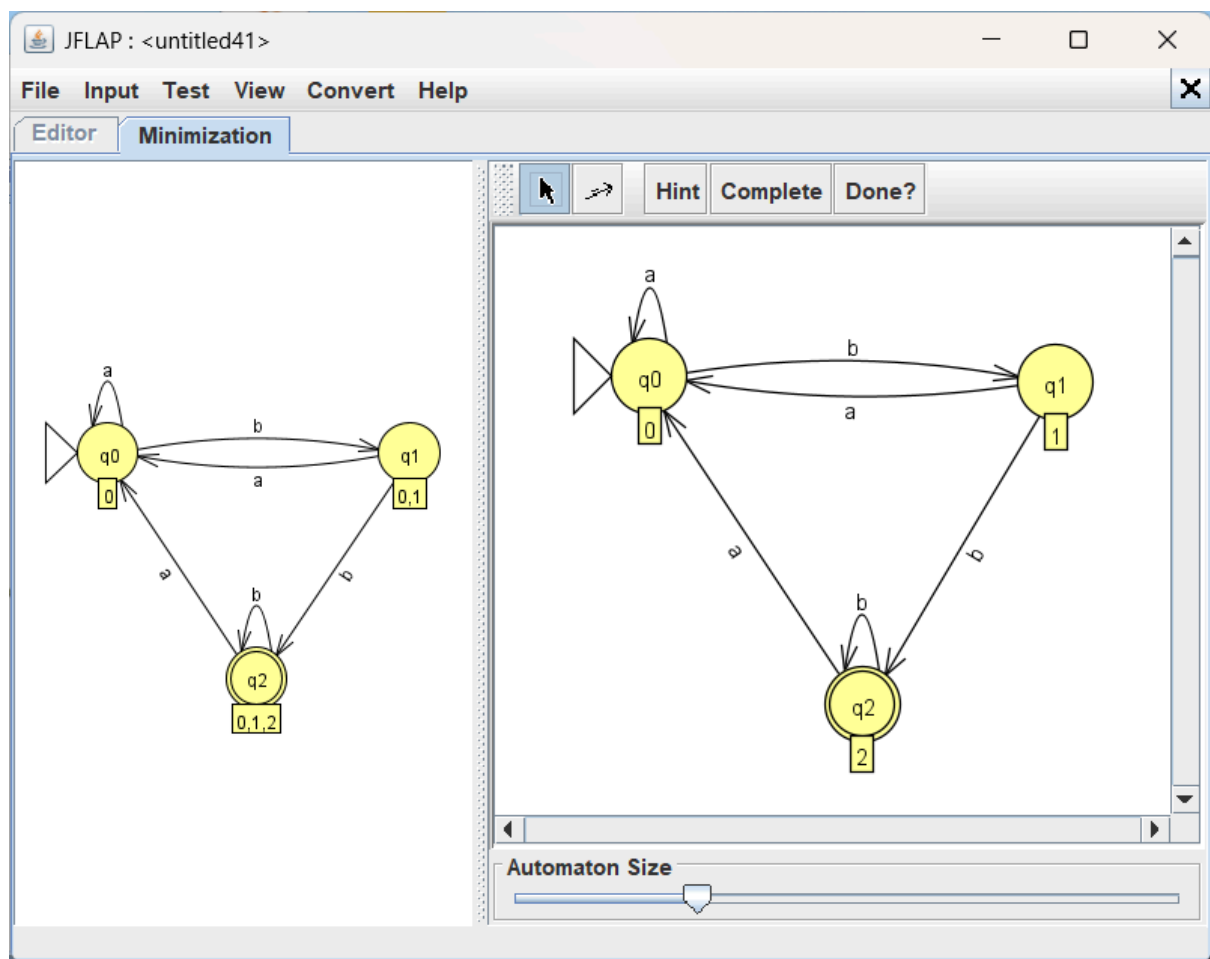


Figura 28: DFA mínimo equivalente.

2.3. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por “a” o terminen en “bb”. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

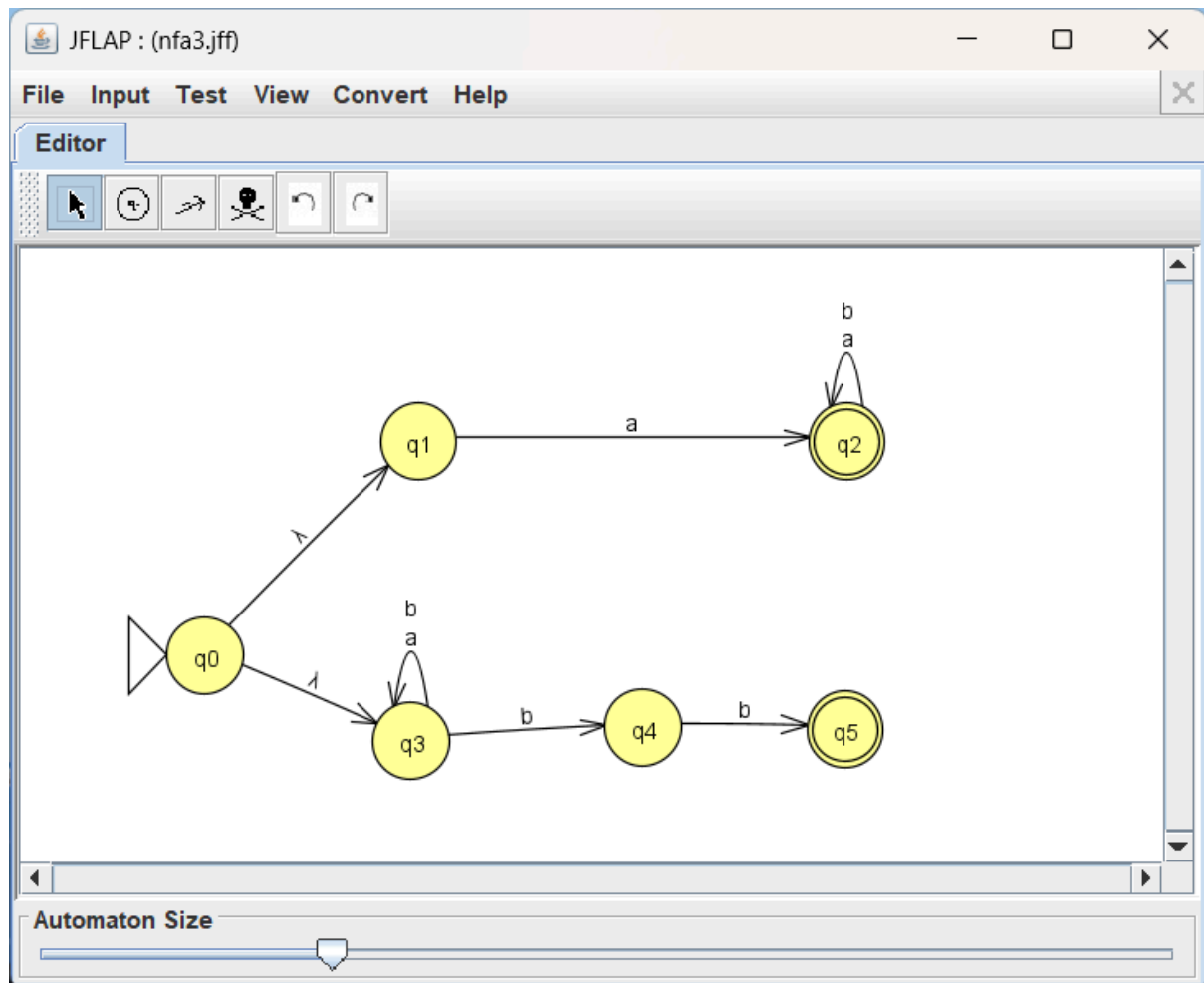


Figura 29: NFA que reconoce cadenas que empiecen por “a” o terminen en “bb” sobre el alfabeto $\Sigma = \{a, b\}$

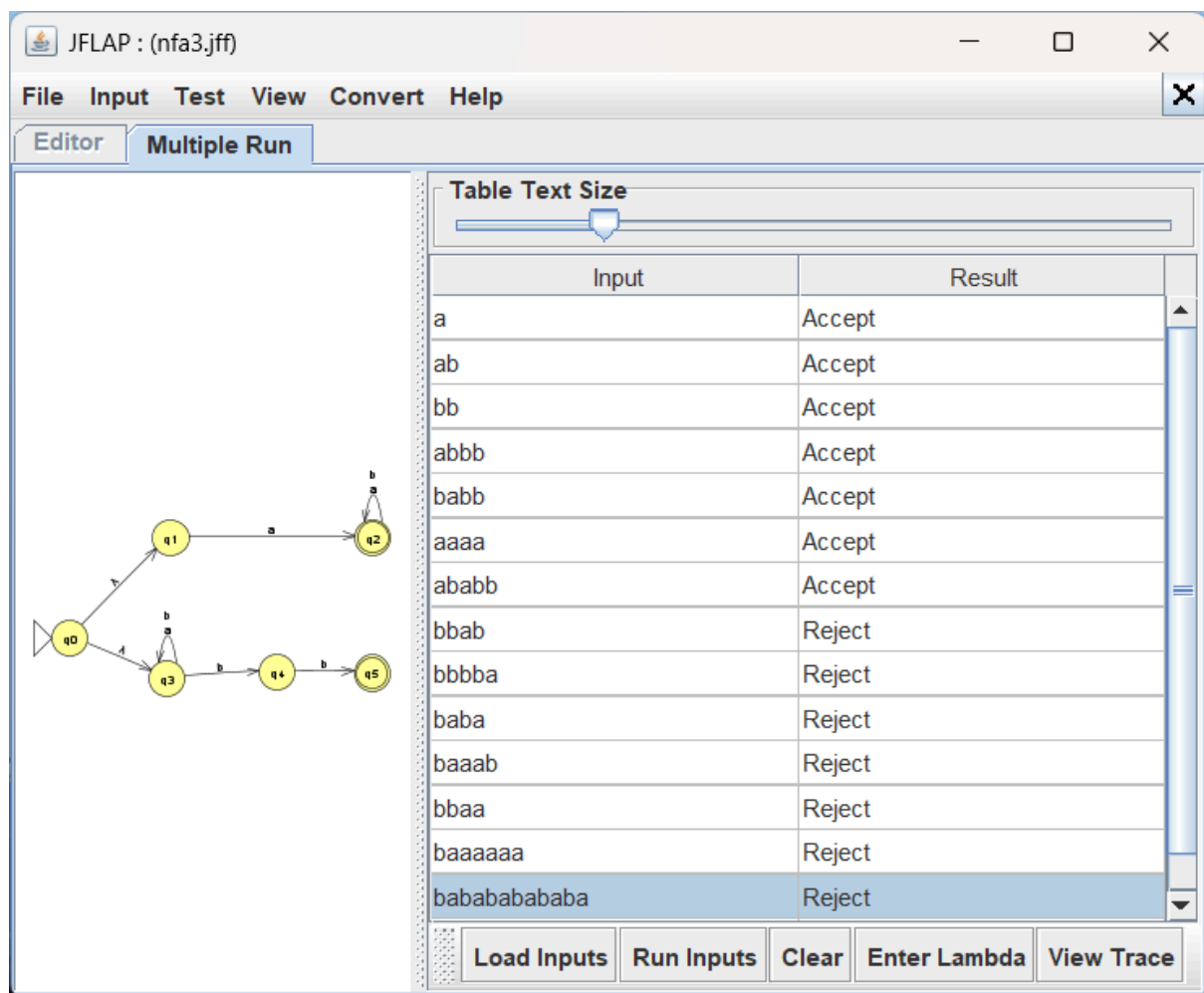


Figura 30: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos para pasar el NFA a un DFA equivalente:

$$\epsilon\text{-clausura}(\{q_0\}) = \{q_0, q_1, q_3\} = A$$

$$\delta(A, a) = \{q_2, q_3\} = B$$

$$\delta(A, b) = \{q_3, q_4\} = C$$

$$\epsilon\text{-clausura}(B) = \{q_2, q_3\} = B$$

$$\delta(B, a) = \{q_2, q_3\} = B$$

$$\delta(B, b) = \{q_2, q_3, q_4\} = D$$

$$\epsilon\text{-clausura}(C) = \{q_3, q_4\} = C$$

$$\delta(C, a) = \{q_3\} = E$$

$$\delta(C, b) = \{q_3, q_4, q_5\} = F$$

$$\epsilon\text{-clausura}(D) = \{q_2, q_3, q_4\} = D$$

$$\delta(D, a) = \{q_2, q_3\} = B$$

$$\delta(D, b) = \{q_2, q_3, q_4, q_5\} = G$$

$$\epsilon\text{-clausura}(E) = \{q_3\} = E$$

$$\delta(E, a) = \{q_3\} = E$$

$\delta(E, b) = \{q3, q4\} = C$
 $\epsilon\text{-clausura}(F) = \{q3, q4, q5\} = F$
 $\delta(F, a) = \{q3\} = E$
 $\delta(F, b) = \{q3, q4, q5\} = F$
 $\epsilon\text{-clausura}(G) = \{q2, q3, q4, q5\} = G$
 $\delta(G, a) = \{q2, q3\} = E$
 $\delta(G, b) = \{q2, q3, q4, q5\} = G$

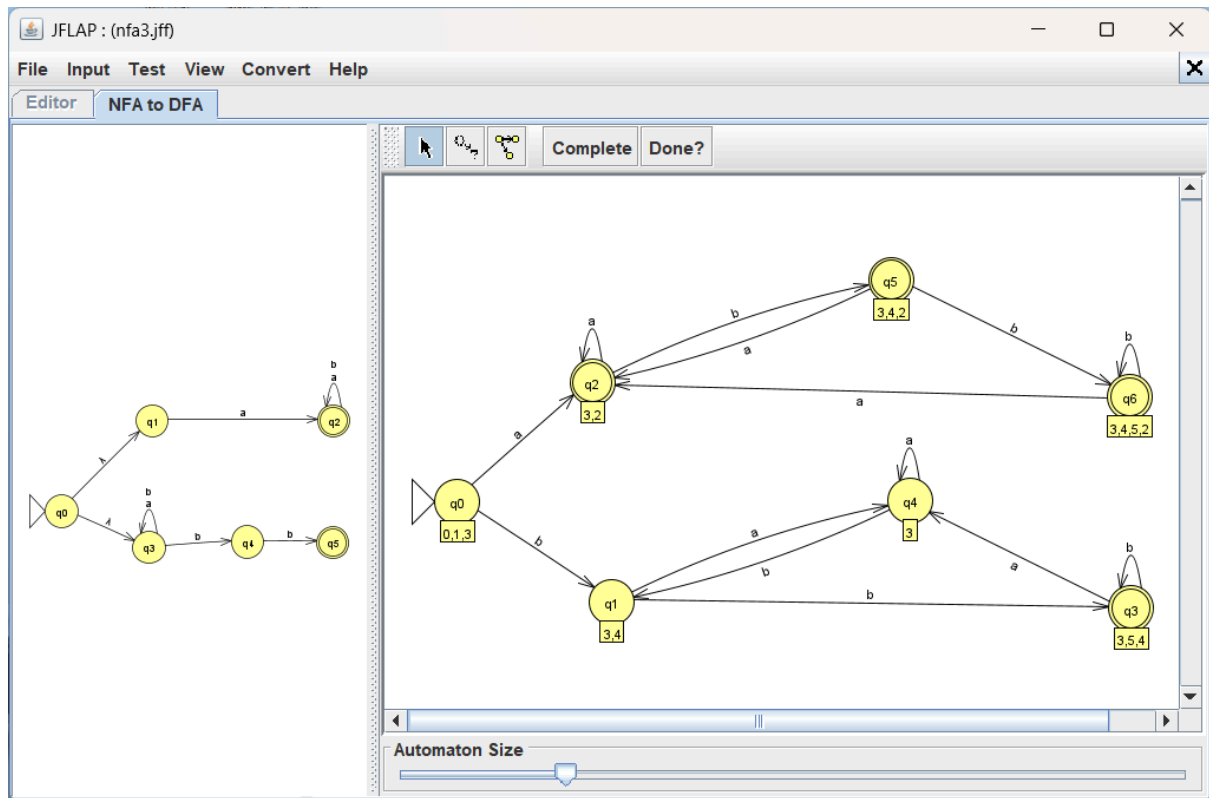


Figura 31: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$\pi = \{\{B, D, G, F\}, \{A, C, E\}\}$
 $\pi' = \{\{B, D, G, F\}, \{A\}, \{C, E\}\}$
 $\pi'' = \{\{B, D, G\}, \{F\}, \{A\}, \{C\}, \{E\}\}$

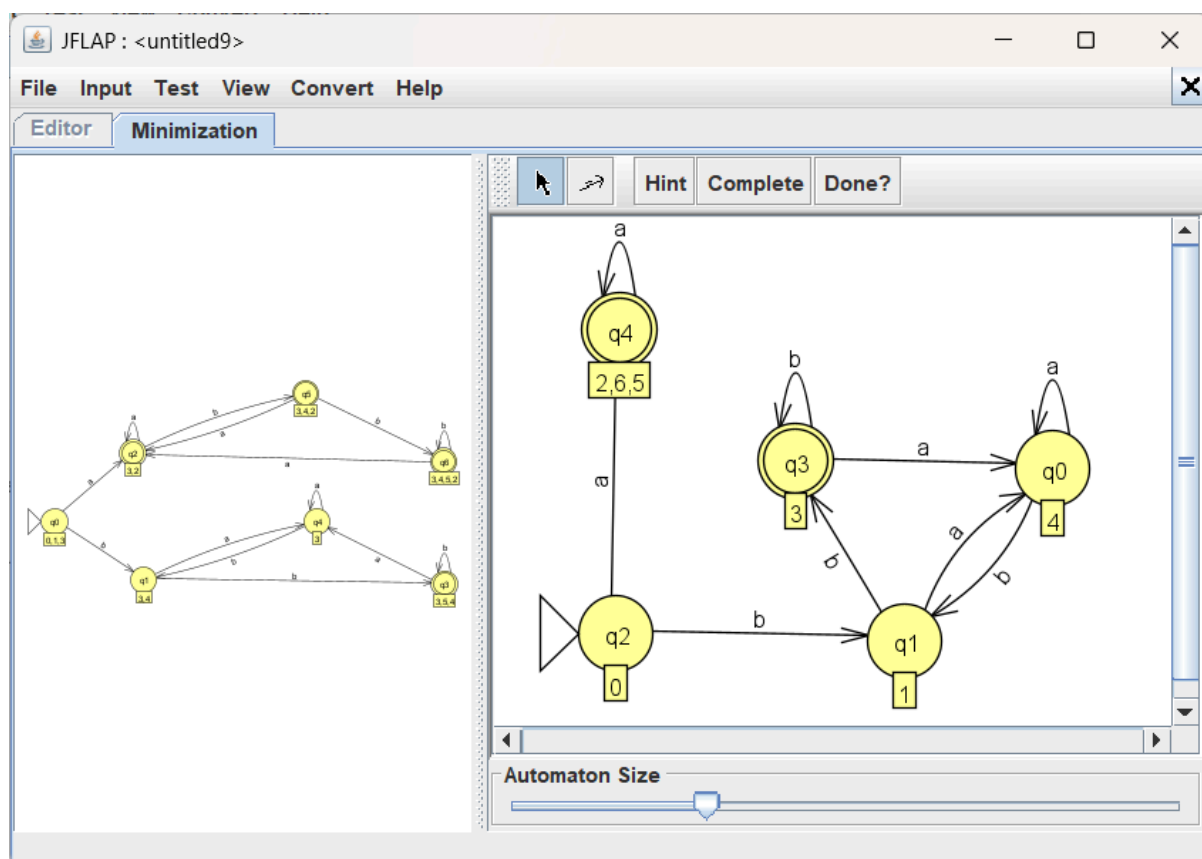


Figura 32: DFA mínimo equivalente.

2.4. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ que empiecen por “a” y terminen en “bb”. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

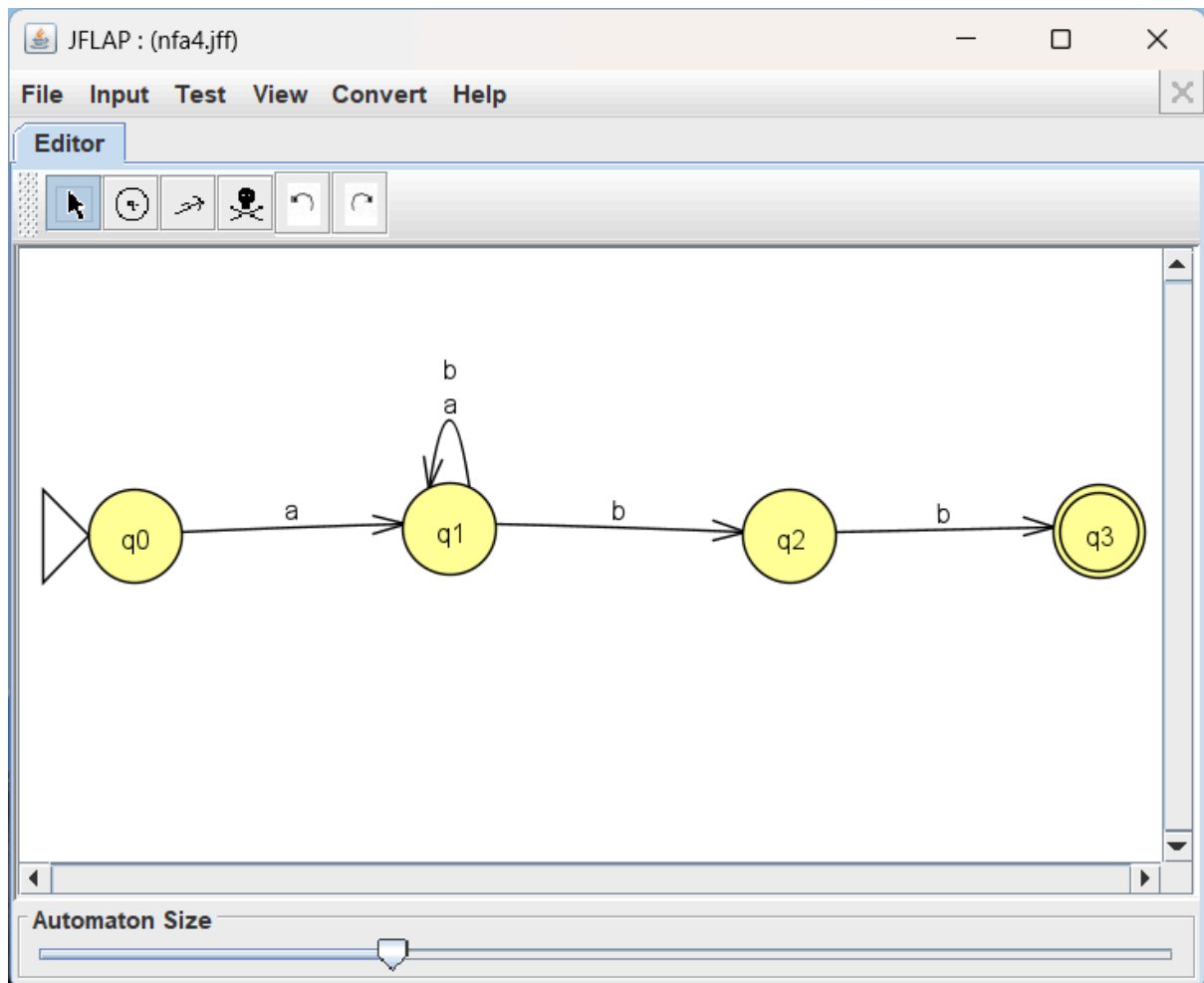


Figura 33: NFA que reconoce cadenas que empiecen por “a” y terminen en “bb” sobre el alfabeto $\Sigma = \{a, b\}$

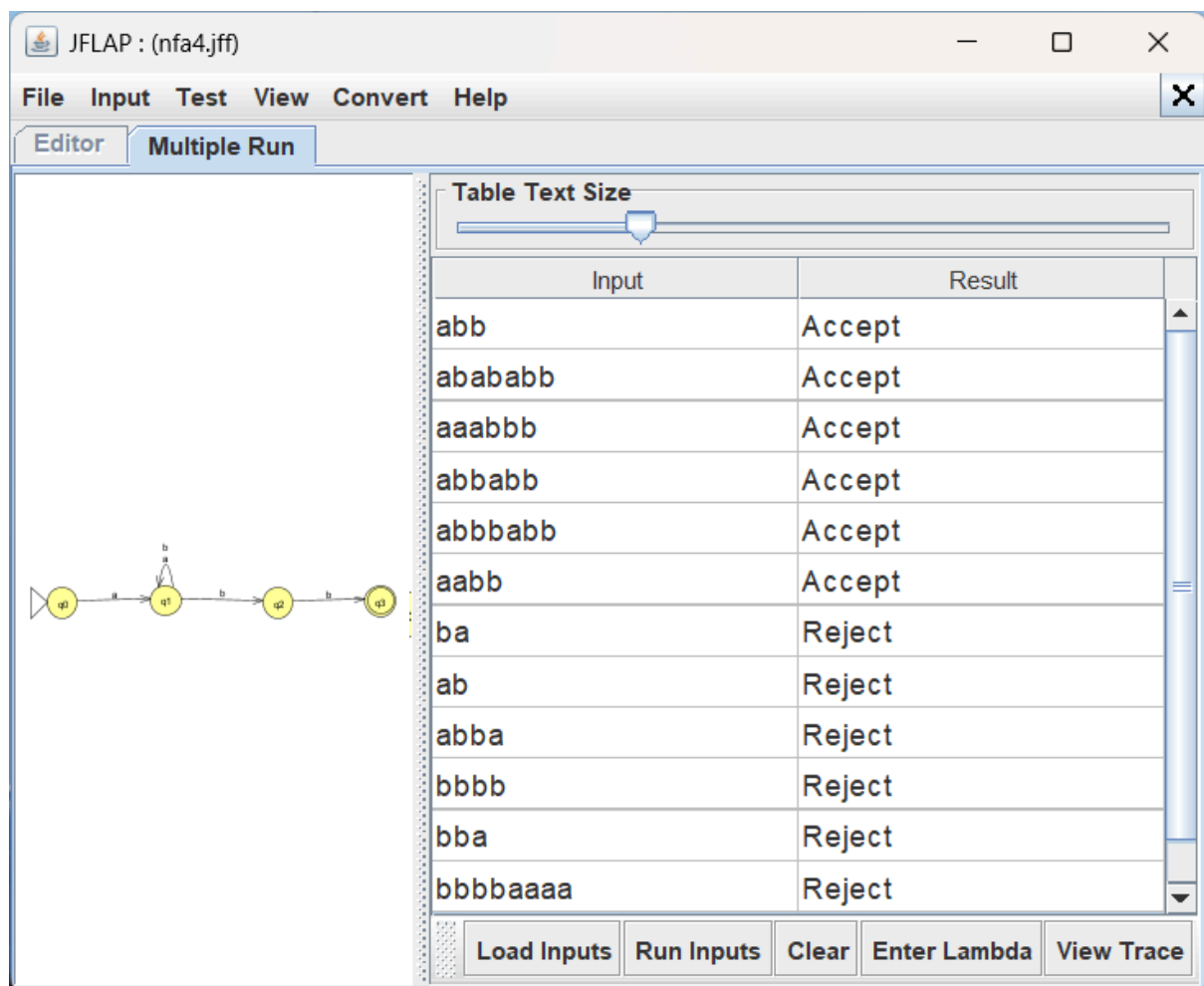


Figura 34: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos (sin ϵ -clausura ya que no hay epsilon transiciones) para pasar el NFA a un DFA equivalente:

$$S = \{q_0\} = A$$

$$\delta(A, a) = \{q_1\} = B$$

$$\delta(A, b) = \emptyset$$

$$\delta(B, a) = \{q_1\} = B$$

$$\delta(B, b) = \{q_1, q_2\} = C$$

$$\delta(C, a) = \{q_1\} = B$$

$$\delta(C, b) = \{q_1, q_2, q_3\} = D$$

$$\delta(D, a) = \{q_1\} = B$$

$$\delta(D, b) = \{q_1, q_2, q_3\} = D$$

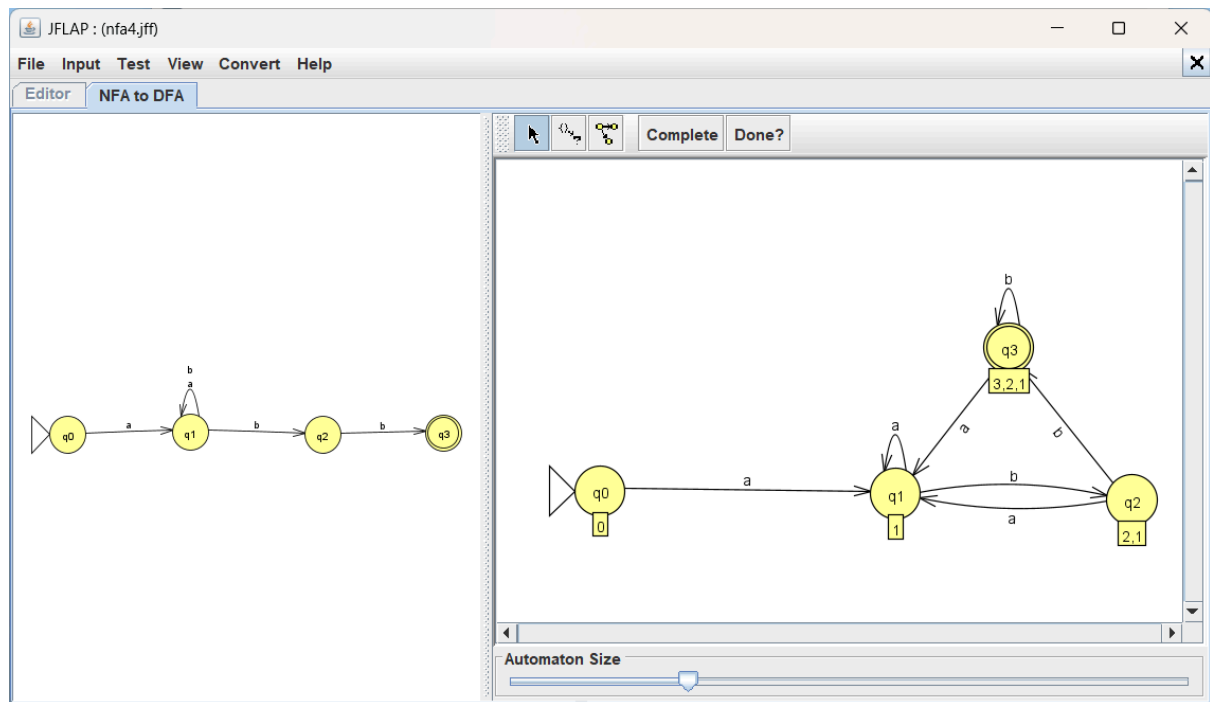


Figura 35: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$$\pi = \{\{D\}, \{A, B, C\}\}$$

$$\pi' = \{\{D\}, \{A\}, \{B, C\}\}$$

$$\pi'' = \{\{D\}, \{A\}, \{B\}, \{C\}\}$$

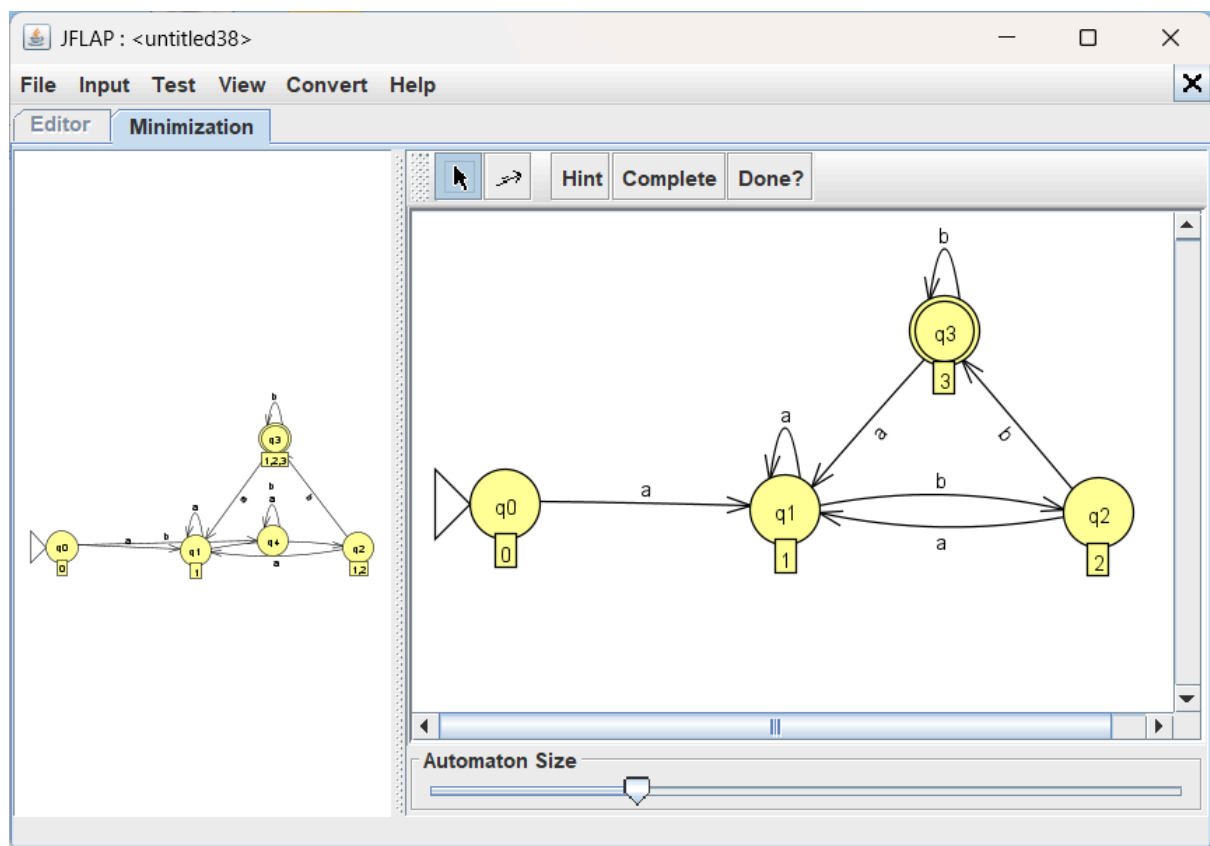


Figura 36: DFA mínimo equivalente.

2.5. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{a, b\}$ con número de “a’s” par o longitud impar. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

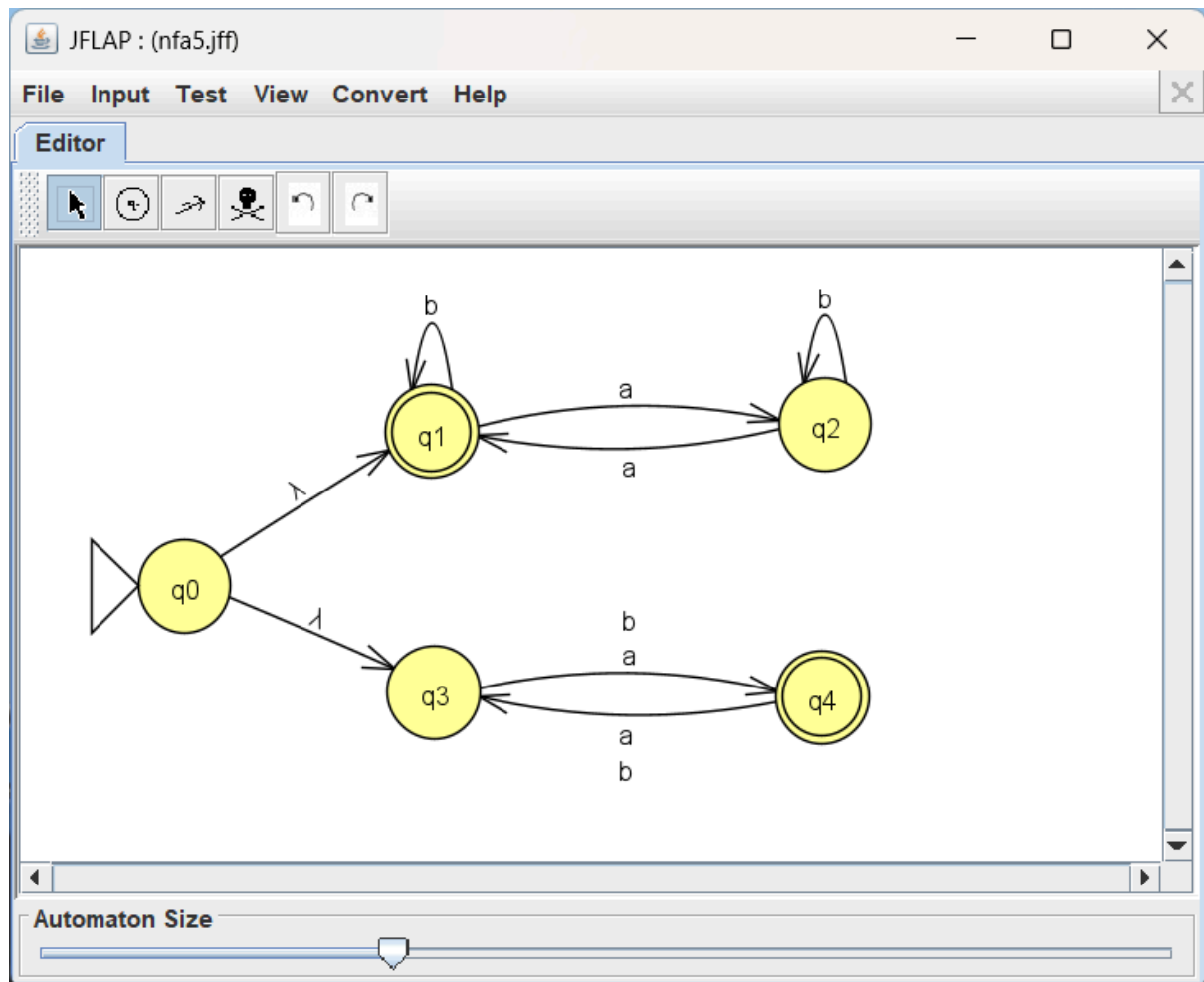


Figura 37: NFA que reconoce cadenas con número de “a’s” par o longitud impar sobre el alfabeto $\Sigma = \{a, b\}$

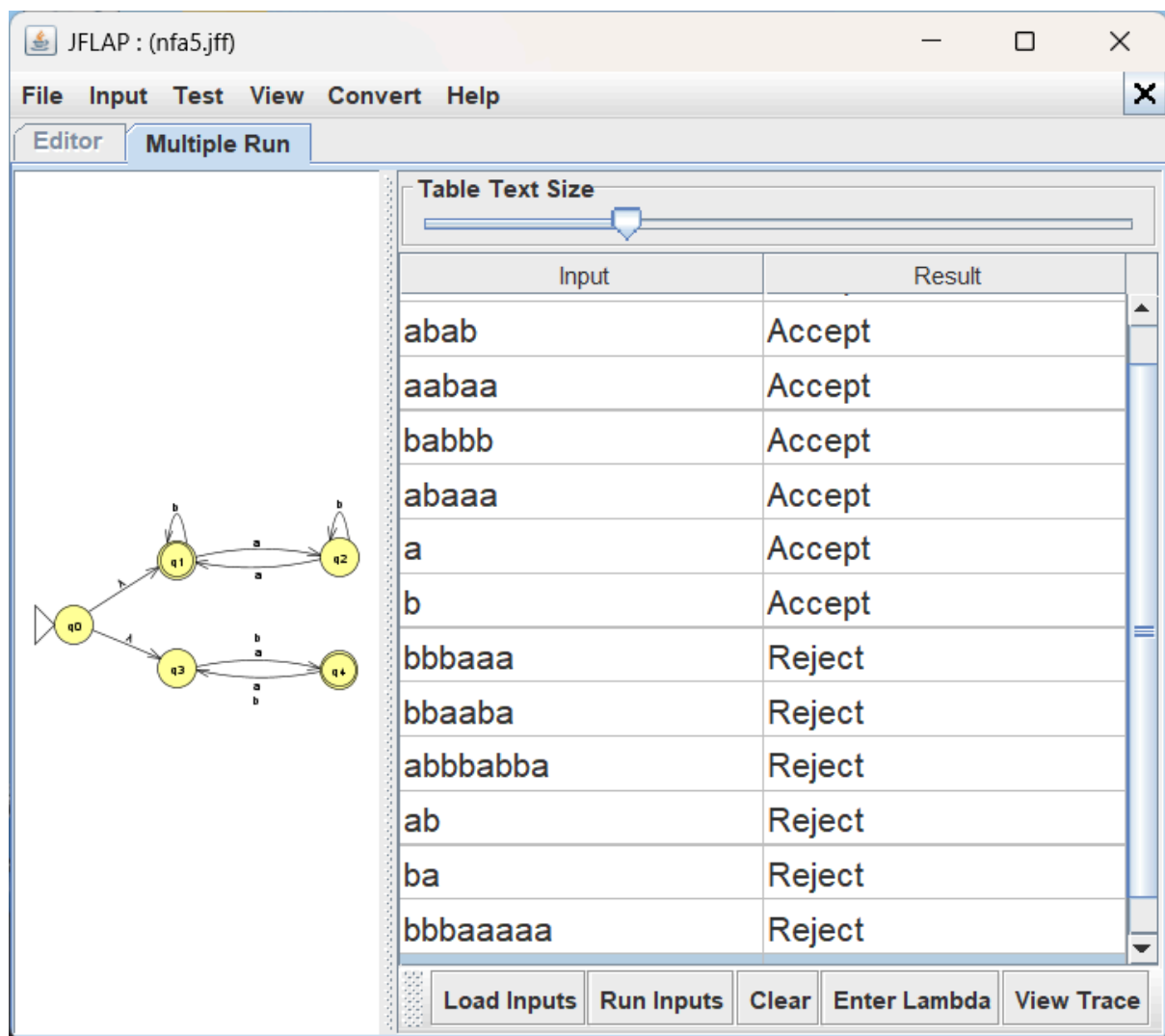


Figura 38: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos para pasar el NFA a un DFA equivalente:

$$\epsilon\text{-clausura}(\{q_0\}) = \{q_0, q_1, q_3\} = A$$

$$\delta(A, a) = \{q_2, q_4\} = B$$

$$\delta(A, b) = \{q_1, q_4\} = C$$

$$\epsilon\text{-clausura}(B) = \{q_2, q_4\} = B$$

$$\delta(B, a) = \{q_1, q_3\} = D$$

$$\delta(B, b) = \{q_2, q_3\} = E$$

$$\epsilon\text{-clausura}(C) = \{q_1, q_4\} = C$$

$$\delta(C, a) = \{q_2, q_3\} = E$$

$$\delta(C, b) = \{q_1, q_3\} = D$$

$$\epsilon\text{-clausura}(D) = \{q_1, q_3\} = D$$

$$\delta(D, a) = \{q_2, q_4\} = B$$

$$\delta(D, b) = \{q_1, q_4\} = C$$

$\epsilon\text{-clausura}(E) = \{q_2, q_3\} = E$

$\delta(E, a) = \{q_1, q_4\} = C$

$\delta(E, b) = \{q_2, q_4\} = B$

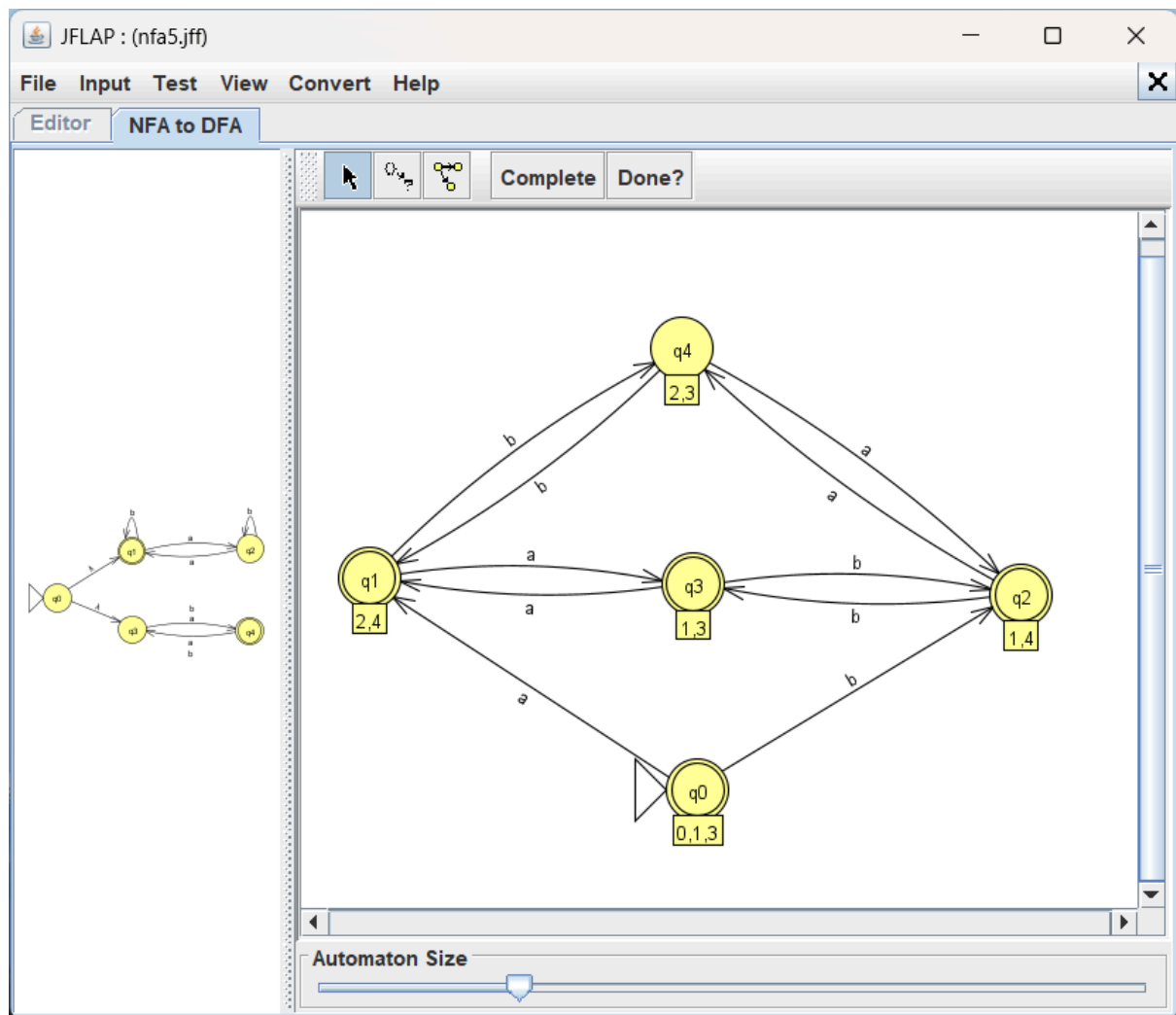


Figura 39: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$\pi = \{\{A, B, C, D\}, \{E\}\}$

$\pi' = \{\{A, B, D\}, \{C\}, \{E\}\}$

$\pi'' = \{\{A\}, \{B, D\}, \{C\}, \{E\}\}$

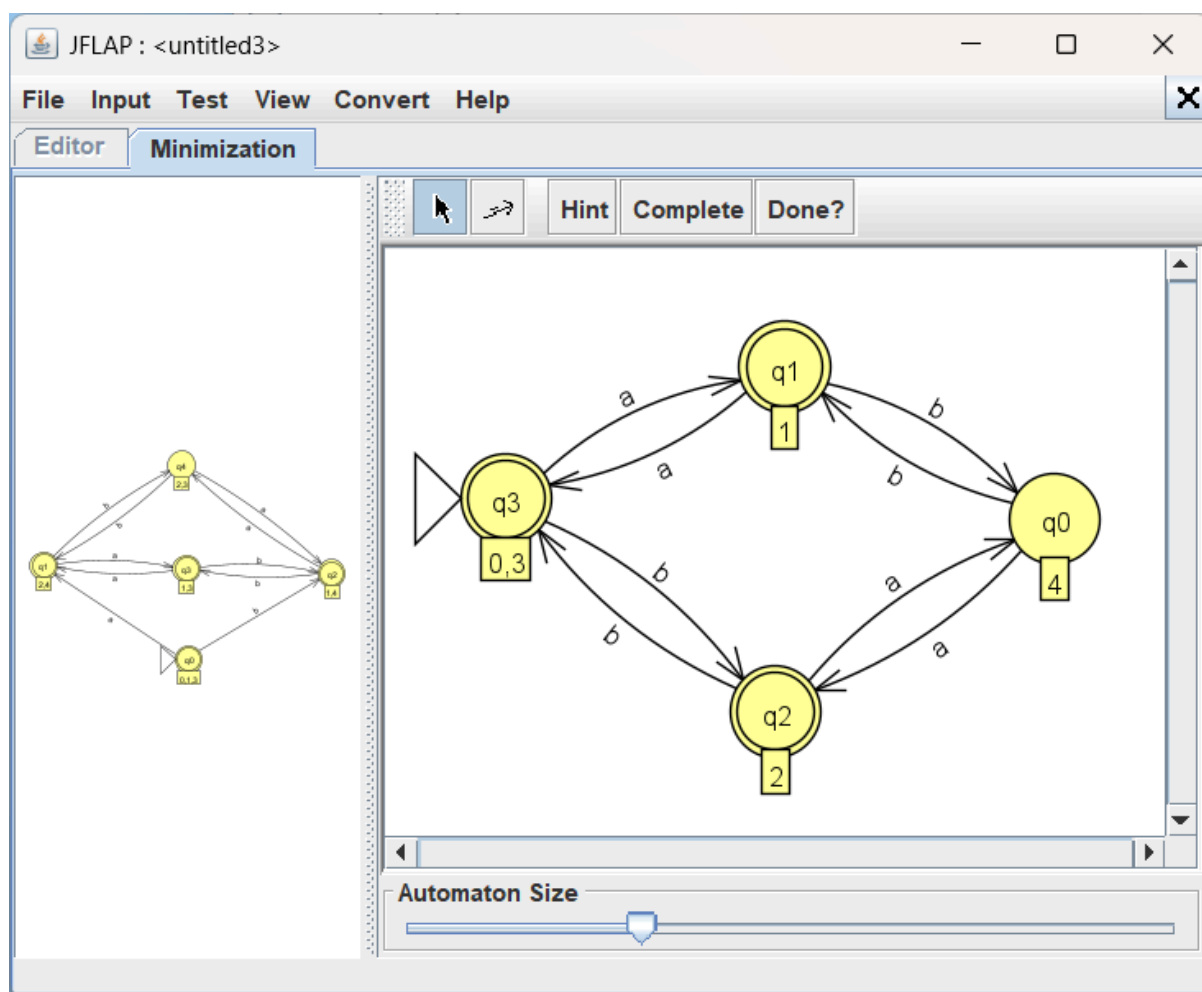


Figura 40: DFA mínimo equivalente.

2.6. Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto $\Sigma = \{x, y, z\}$ que contengan al menos dos símbolos iguales consecutivos. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

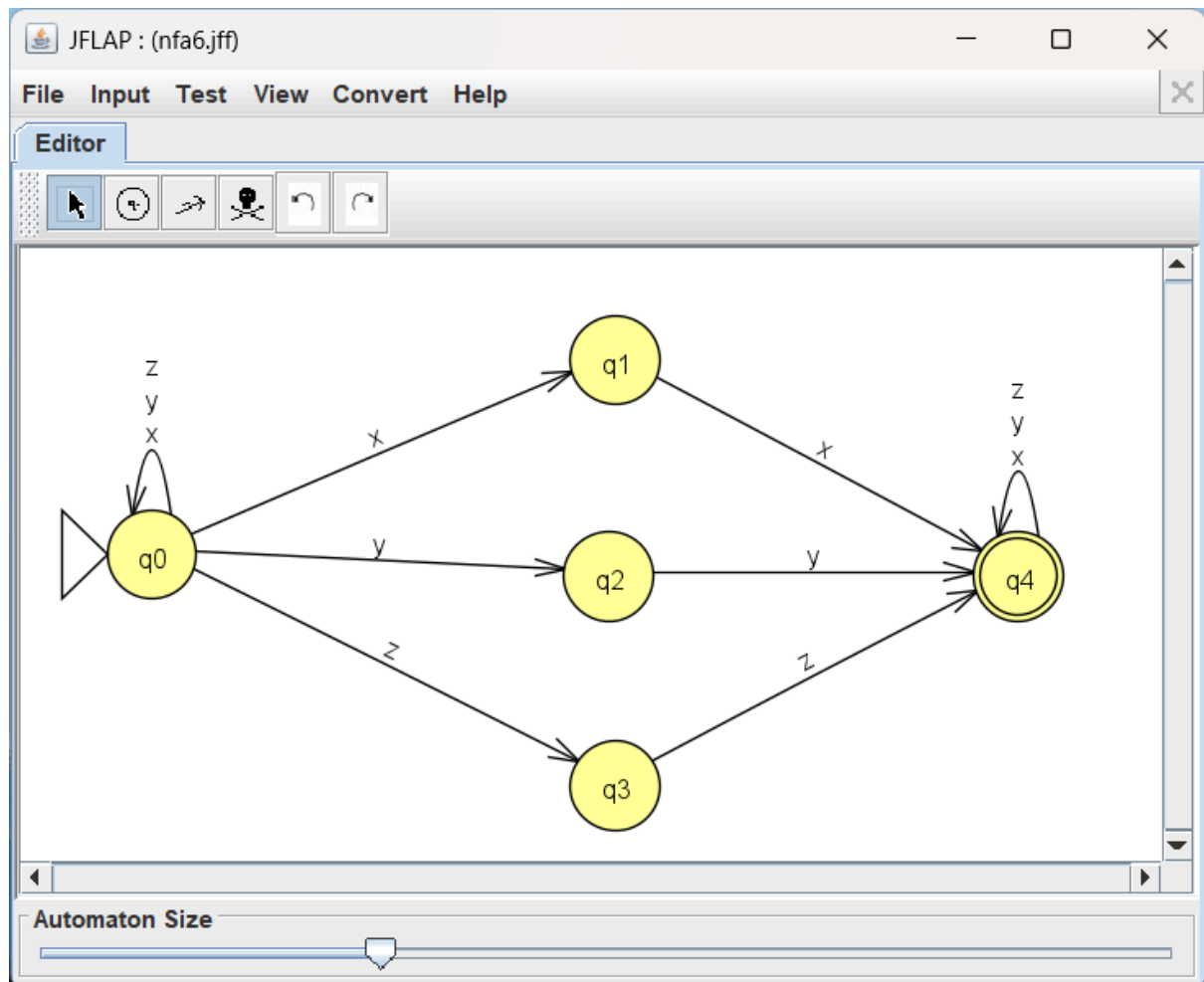


Figura 41: NFA que reconoce cadenas que contengan al menos dos símbolos iguales consecutivos sobre el alfabeto $\Sigma = \{x, y, z\}$

JFLAP : (nfa6.jff)

File Input Test View Convert Help

Editor Multiple Run

Table Text Size

Input	Result
xyz	Reject
xyzx	Reject
zxzyzx	Reject
zxyxz	Reject
zyxyzyxyzyxyz	Reject
xyxxz	Accept
zxyxx	Accept
zz	Accept
yyyyxx	Accept
xyxyzz	Accept

Load Inputs Run Inputs Clear Enter Lambda View Trace

Figura 42: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos (sin ϵ -clausura ya que no hay epsilon transiciones) para pasar el NFA a un DFA equivalente:

$S = \{q_0\} = A$
 $\delta(A, x) = \{q_0, q_1\} = B$
 $\delta(A, y) = \{q_0, q_2\} = C$
 $\delta(A, z) = \{q_0, q_3\} = D$
 $\delta(B, x) = \{q_0, q_1, q_4\} = E$
 $\delta(B, y) = \{q_0, q_2\} = C$
 $\delta(B, z) = \{q_0, q_3\} = D$
 $\delta(C, x) = \{q_0, q_1\} = B$
 $\delta(C, y) = \{q_0, q_2, q_4\} = F$
 $\delta(C, z) = \{q_0, q_3\} = D$
 $\delta(D, x) = \{q_0, q_1\} = B$
 $\delta(D, y) = \{q_0, q_2\} = C$
 $\delta(D, z) = \{q_0, q_3, q_4\} = G$
 $\delta(E, x) = \{q_0, q_1, q_4\} = E$
 $\delta(E, y) = \{q_0, q_2, q_4\} = F$

$\delta(E, z) = \{q_0, q_3, q_4\} = G$
 $\delta(F, x) = \{q_0, q_1, q_4\} = E$
 $\delta(F, y) = \{q_0, q_2, q_4\} = F$
 $\delta(F, z) = \{q_0, q_3, q_4\} = G$
 $\delta(G, x) = \{q_0, q_1, q_4\} = E$
 $\delta(G, y) = \{q_0, q_2, q_4\} = F$
 $\delta(G, z) = \{q_0, q_3, q_4\} = G$

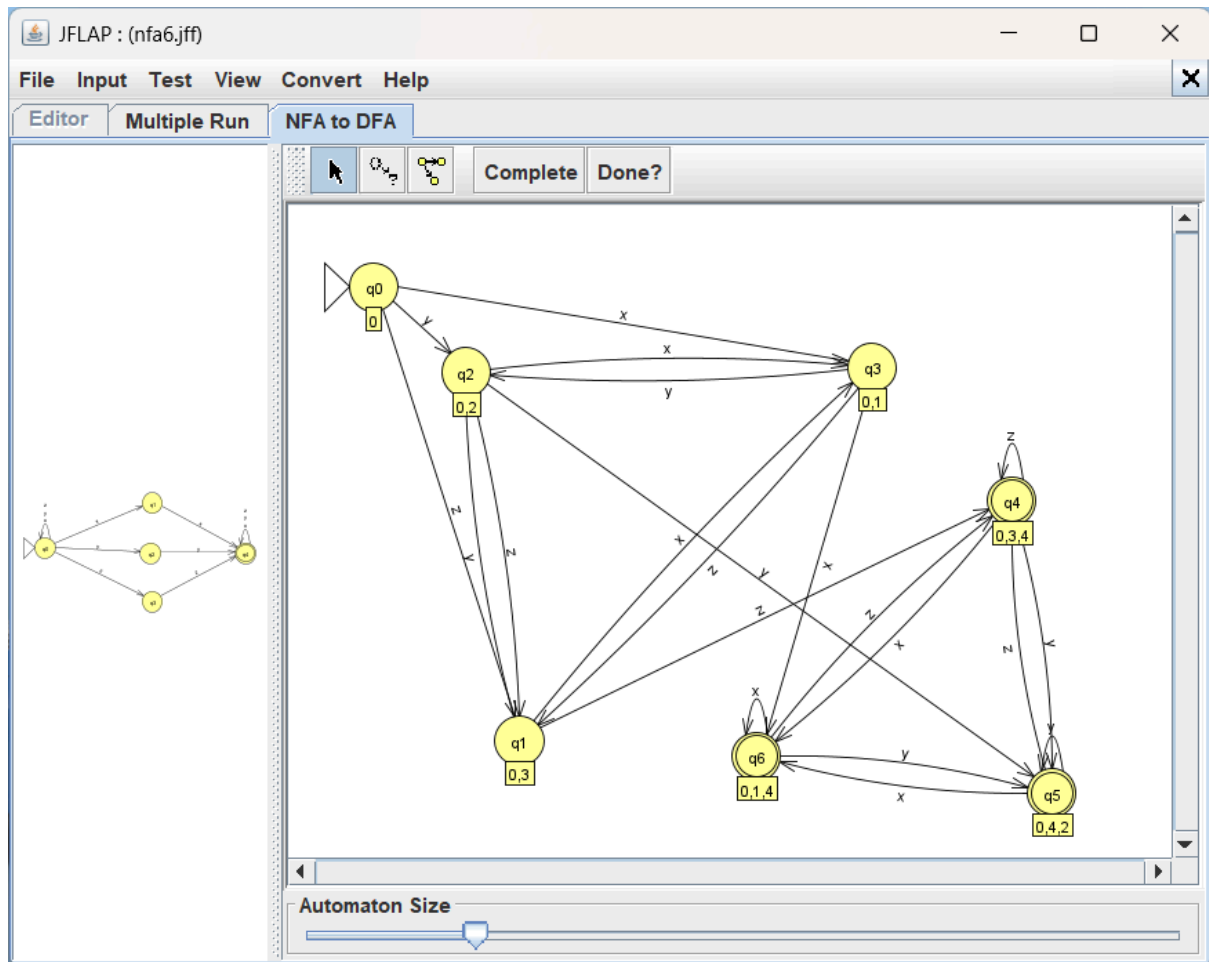


Figura 43: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$\pi = \{\{E, F, G\}, \{A, B, C, D\}\}$
 $\pi' = \{\{E, F, G\}, \{B\}, \{A, C, D\}\}$
 $\pi'' = \{\{E, F, G\}, \{B\}, \{C\}, \{A, D\}\}$
 $\pi''' = \{\{E, F, G\}, \{B\}, \{C\}, \{D\}, \{A\}\}$

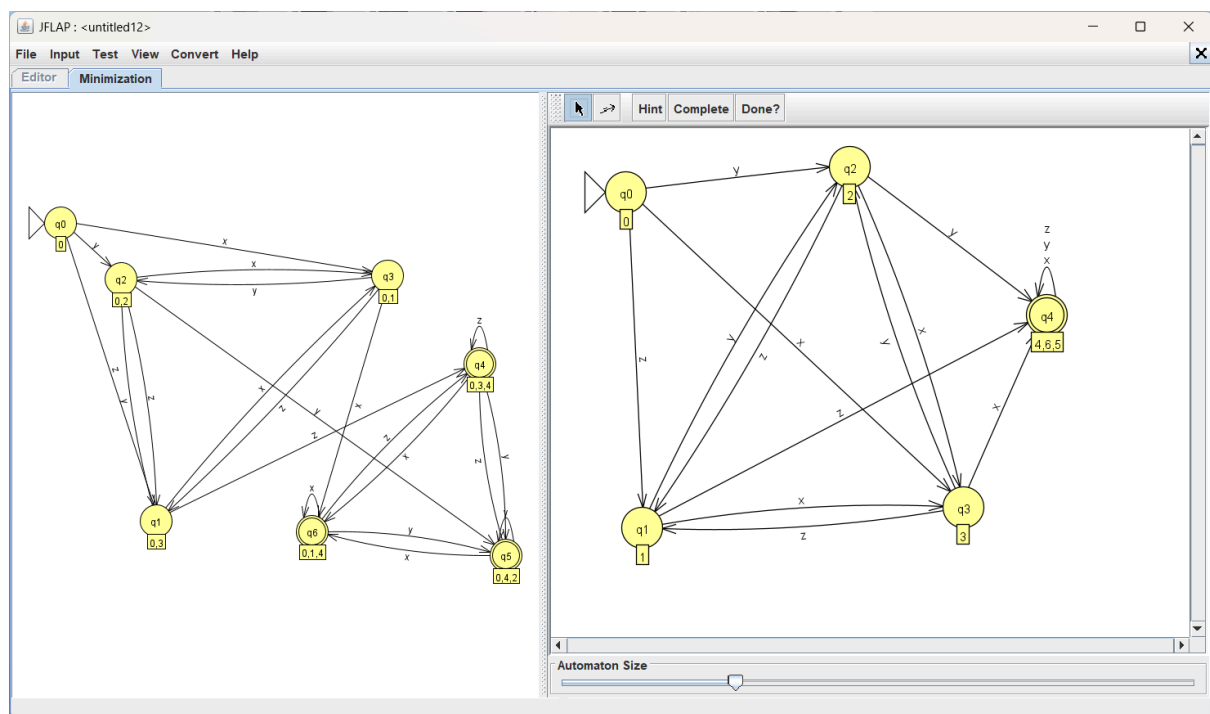


Figura 44: DFA mínimo equivalente.

2.7. Diseñar un autómata finito no determinista que reconozca cadenas w sobre el alfabeto $\{x, y, z\}$, con $|w| \geq 2$, tales que w empieza y termina por el mismo símbolo. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

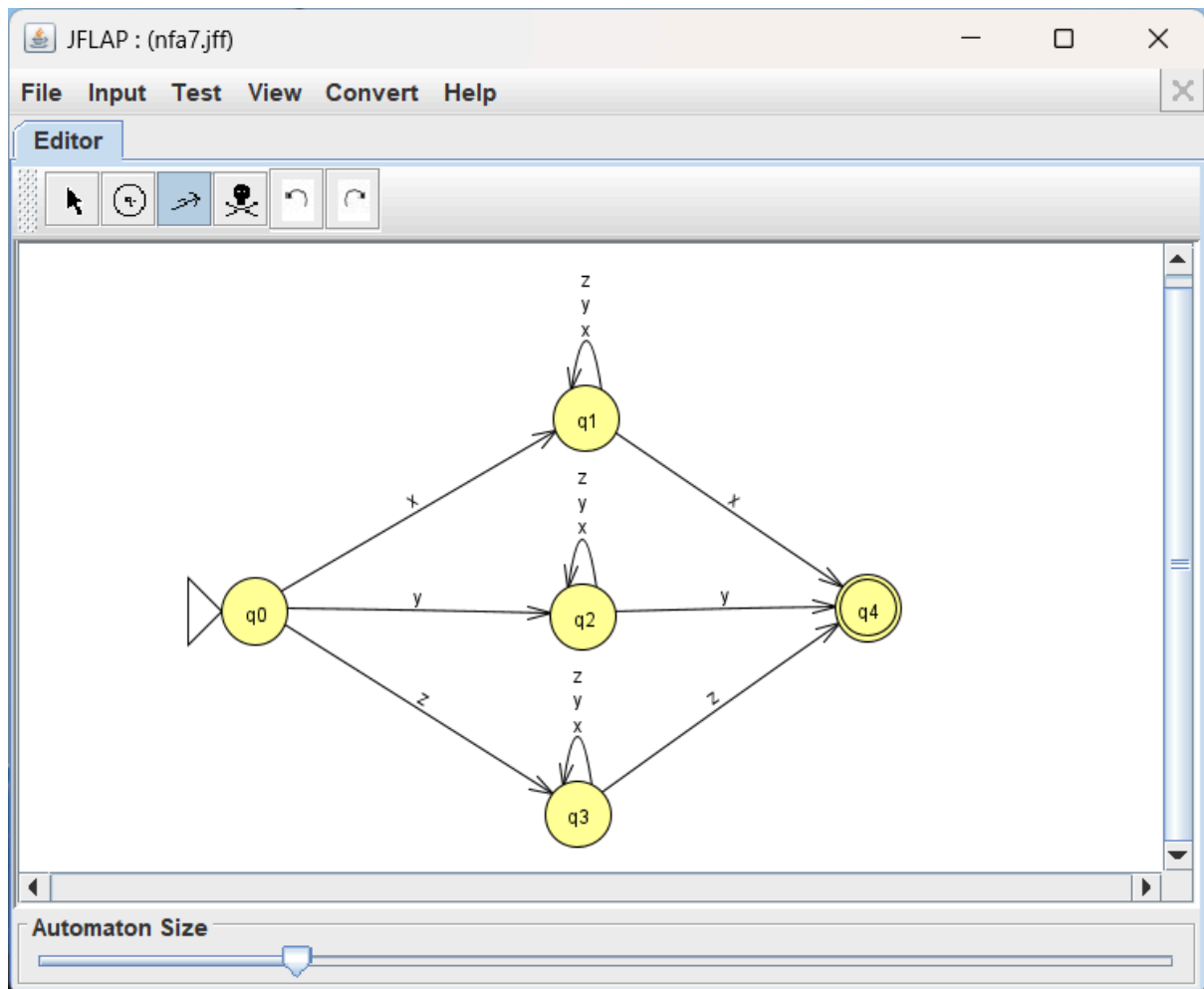


Figura 45: NFA que reconoce cadenas w sobre el alfabeto $\{x, y, z\}$, con $|w| \geq 2$, tales que w empieza y termina por el mismo símbolo.

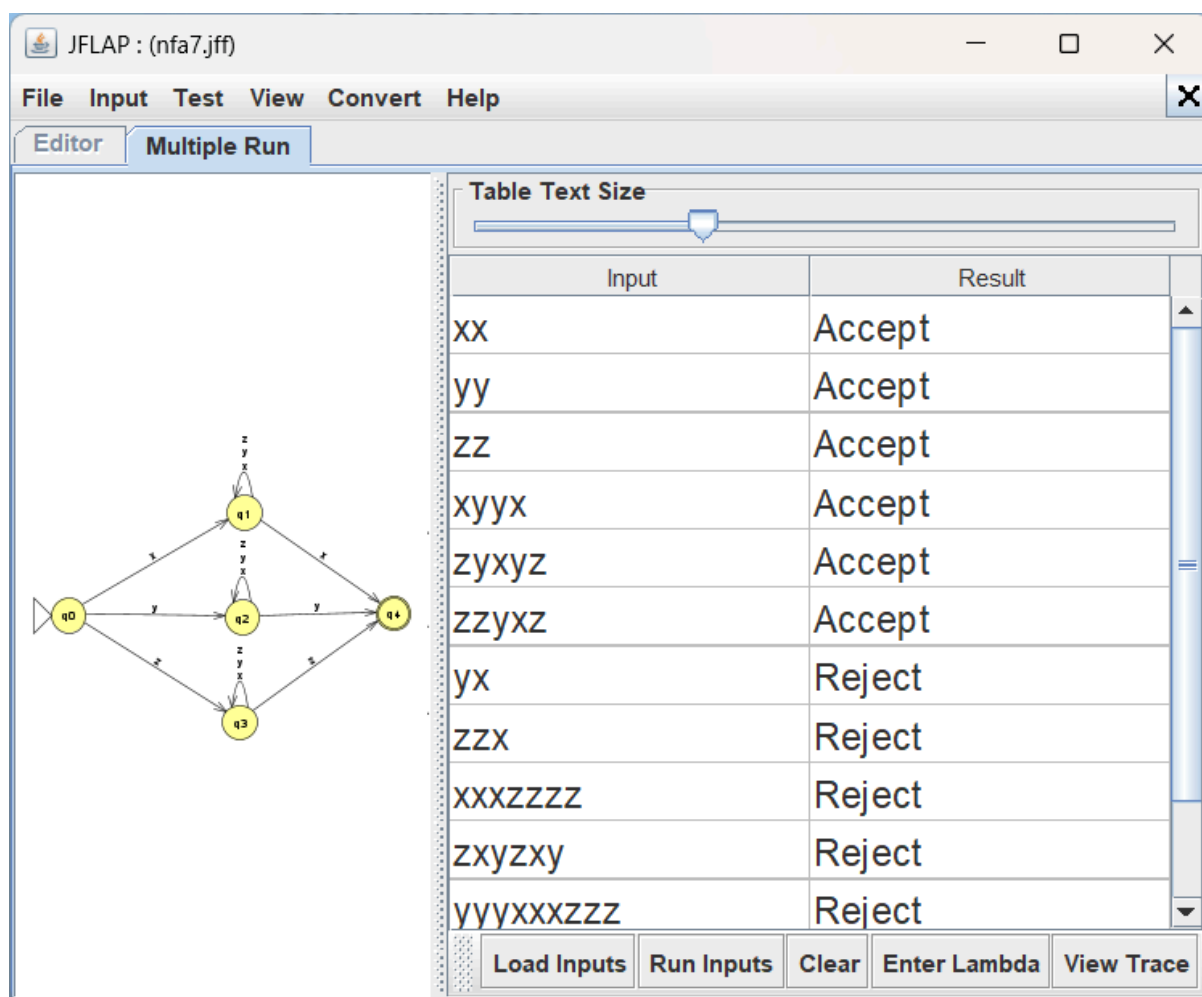


Figura 46: Cadenas de prueba para el NFA anterior.

Algoritmo de Construcción de Subconjuntos (sin ϵ -clausura ya que no hay epsilon transiciones) para pasar el NFA a un DFA equivalente:

$$S = \{q_0\} = A$$

$$\delta(A, x) = \{q_1\} = B$$

$$\delta(A, y) = \{q_2\} = C$$

$$\delta(A, z) = \{q_3\} = D$$

$$\delta(B, x) = \{q_1, q_4\} = E$$

$$\delta(B, y) = \{q_1\} = B$$

$$\delta(B, z) = \{q_1\} = B$$

$$\delta(C, x) = \{q_2\} = C$$

$$\delta(C, y) = \{q_2, q_4\} = F$$

$$\delta(C, z) = \{q_2\} = C$$

$$\delta(D, x) = \{q_3\} = D$$

$$\delta(D, y) = \{q_3\} = D$$

$$\delta(D, z) = \{q_3, q_4\} = G$$

$$\delta(E, x) = \{q_1, q_4\} = E$$

$\delta(E, y) = \{q1\} = B$
 $\delta(E, z) = \{q1\} = B$
 $\delta(F, x) = \{q2\} = C$
 $\delta(F, y) = \{q2, q4\} = F$
 $\delta(F, z) = \{q2\} = C$
 $\delta(G, x) = \{q3\} = D$
 $\delta(G, y) = \{q3\} = D$
 $\delta(G, z) = \{q3, q4\} = G$

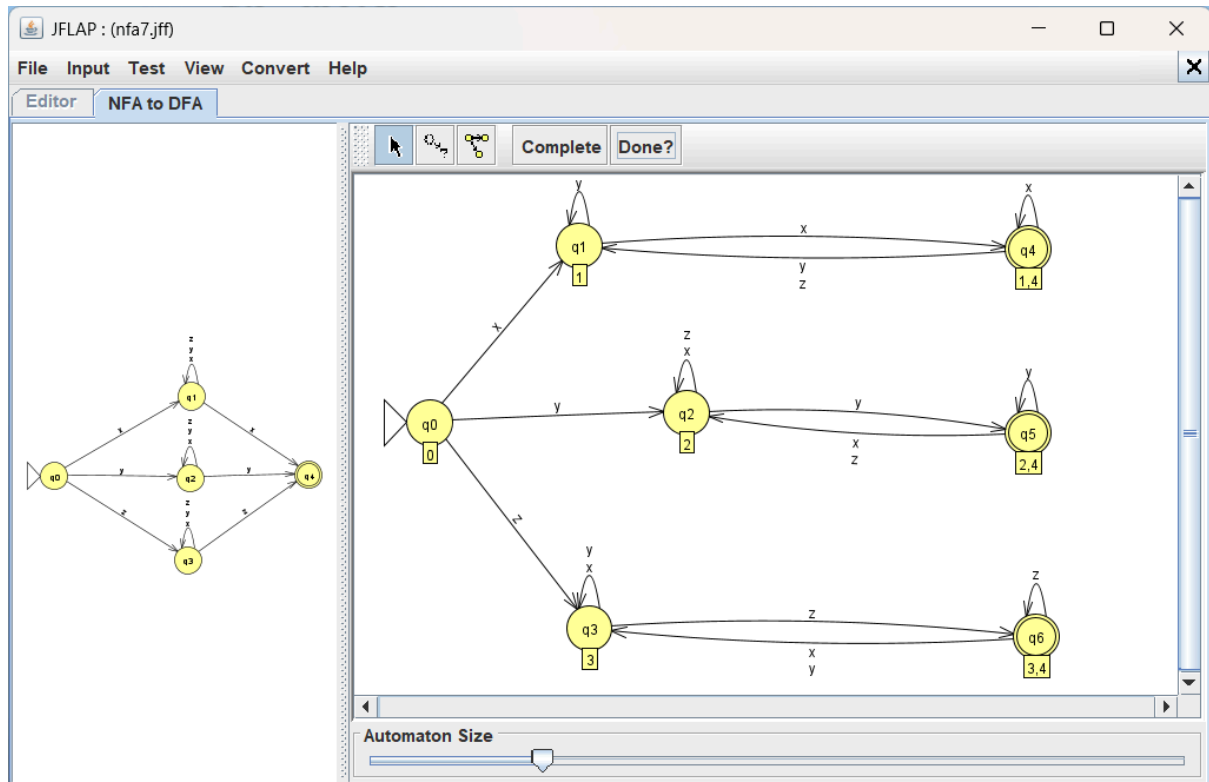


Figura 47: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

$\pi = \{\{E, F, G\}, \{A, B, C, D\}\}$
 $\pi' = \{\{E, F, G\}, \{B\}, \{A, C, D\}\}$
 $\pi'' = \{\{E, F, G\}, \{B\}, \{C\}, \{A, D\}\}$
 $\pi''' = \{\{E, F, G\}, \{B\}, \{C\}, \{D\}, \{A\}\}$
 $\pi'''' = \{\{E, G\}, \{F\}, \{B\}, \{C\}, \{D\}, \{A\}\}$
 $\pi''''' = \{\{E\}, \{G\}, \{F\}, \{B\}, \{C\}, \{D\}, \{A\}\}$

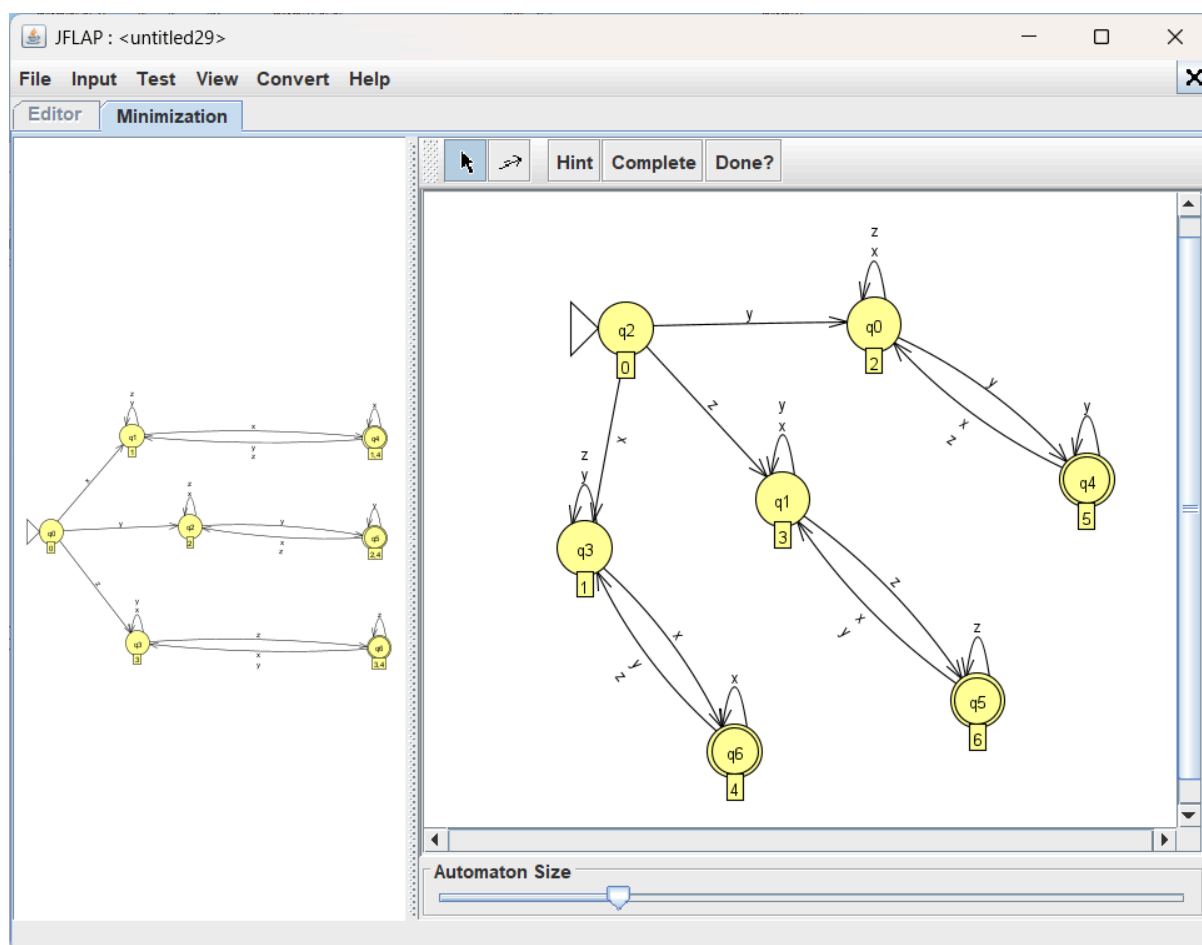


Figura 48: DFA mínimo equivalente.

3. Modificación

3.1. Diseñar un NFA que reconozca las cadenas del lenguaje binario con dos ceros consecutivos y dos unos consecutivos.

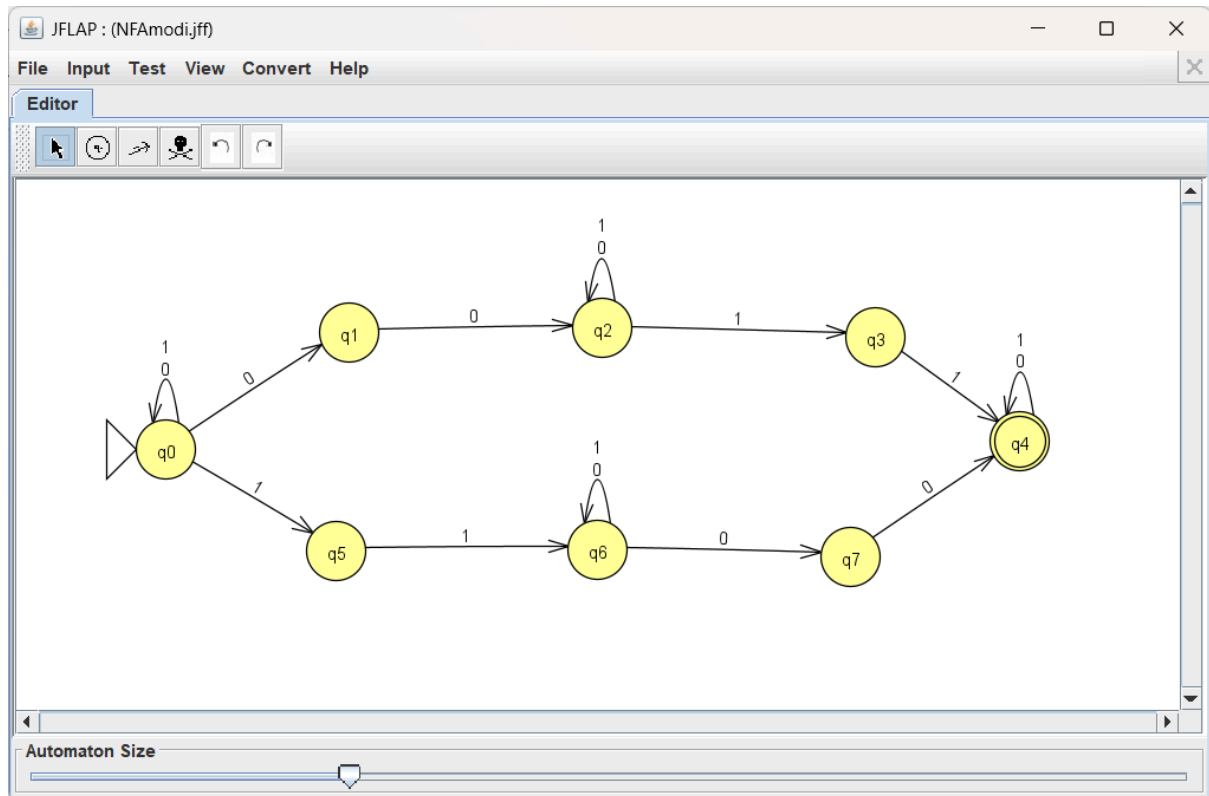


Figura 49: del lenguaje binario con dos ceros consecutivos y dos unos consecutivos.

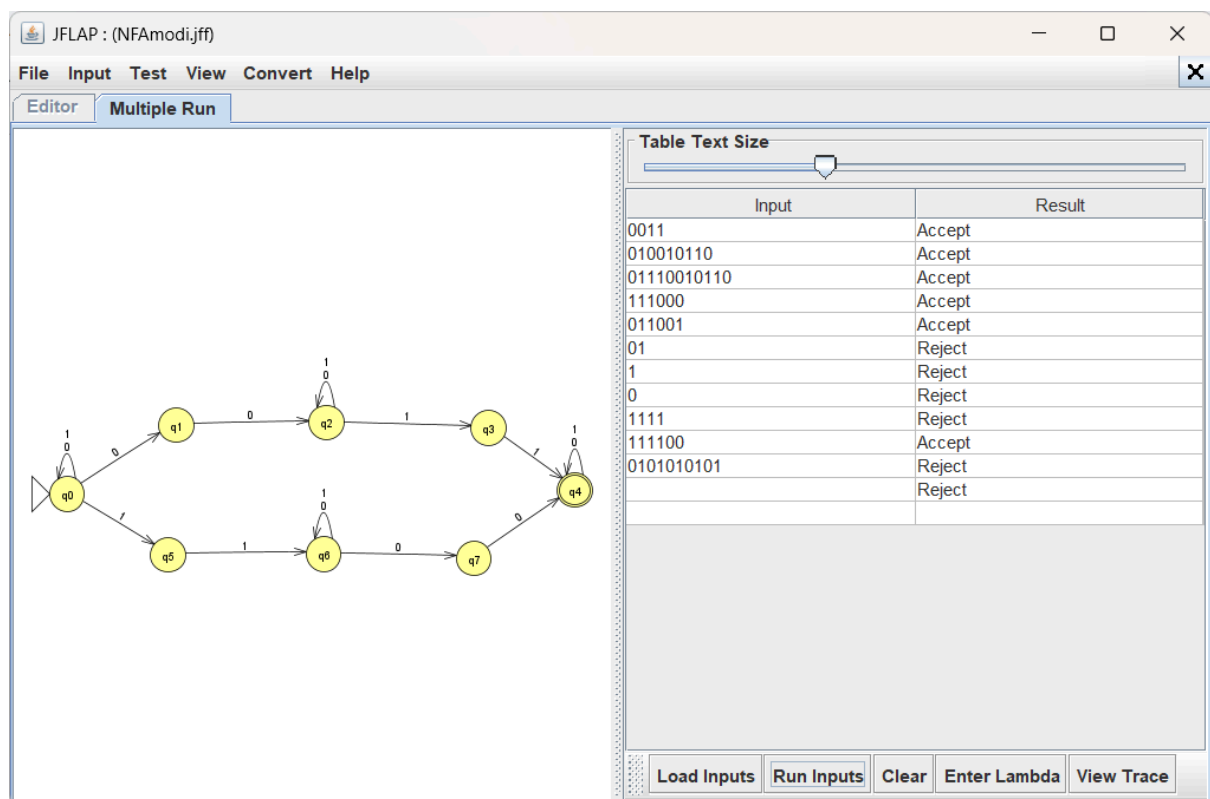


Figura 50: Cadenas de prueba para el NFA anterior.

Utilizamos el JFLAP para obtener un DFA equivalente:

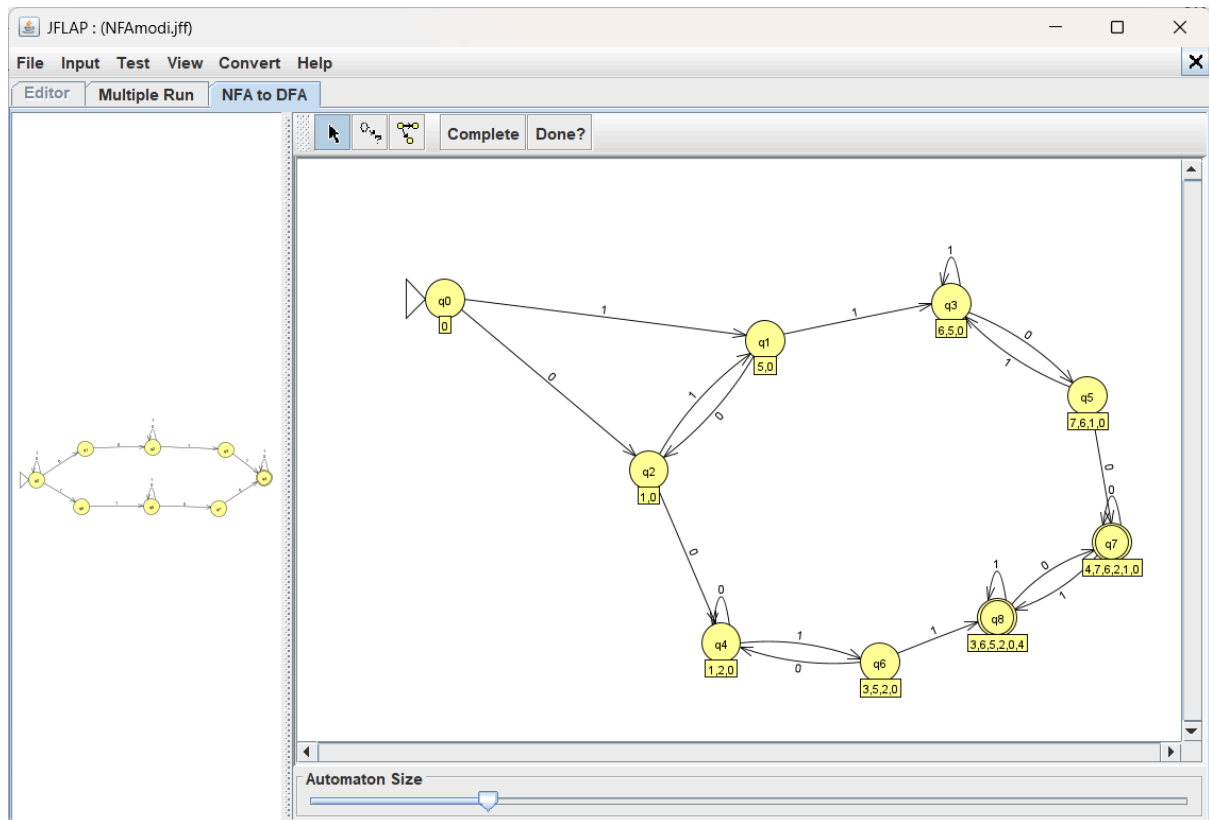


Figura 51: DFA equivalente al NFA anterior.

Utilizamos el JFLAP para minimizar el DFA anterior:

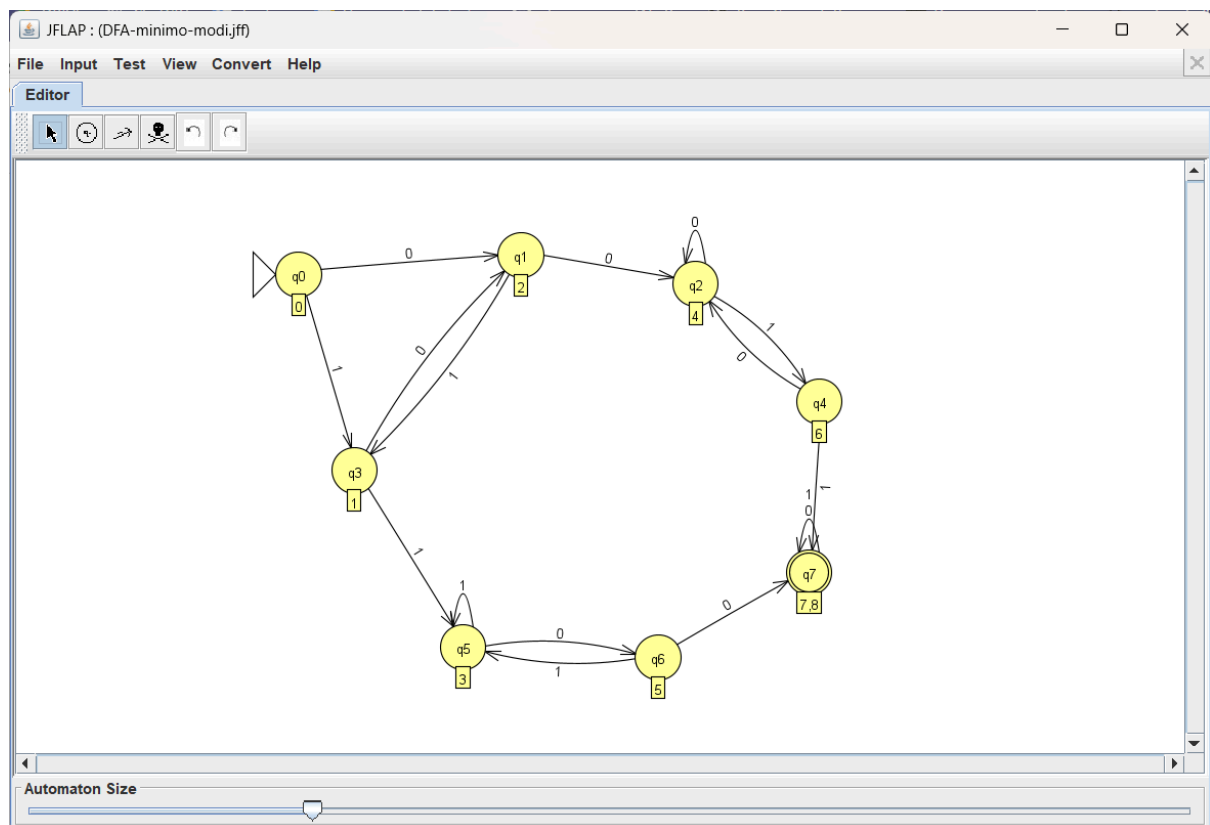


Figura 52: DFA mínimo equivalente.

Comprobamos la equivalencia del NFA y el DFA mediante el JFLAP:

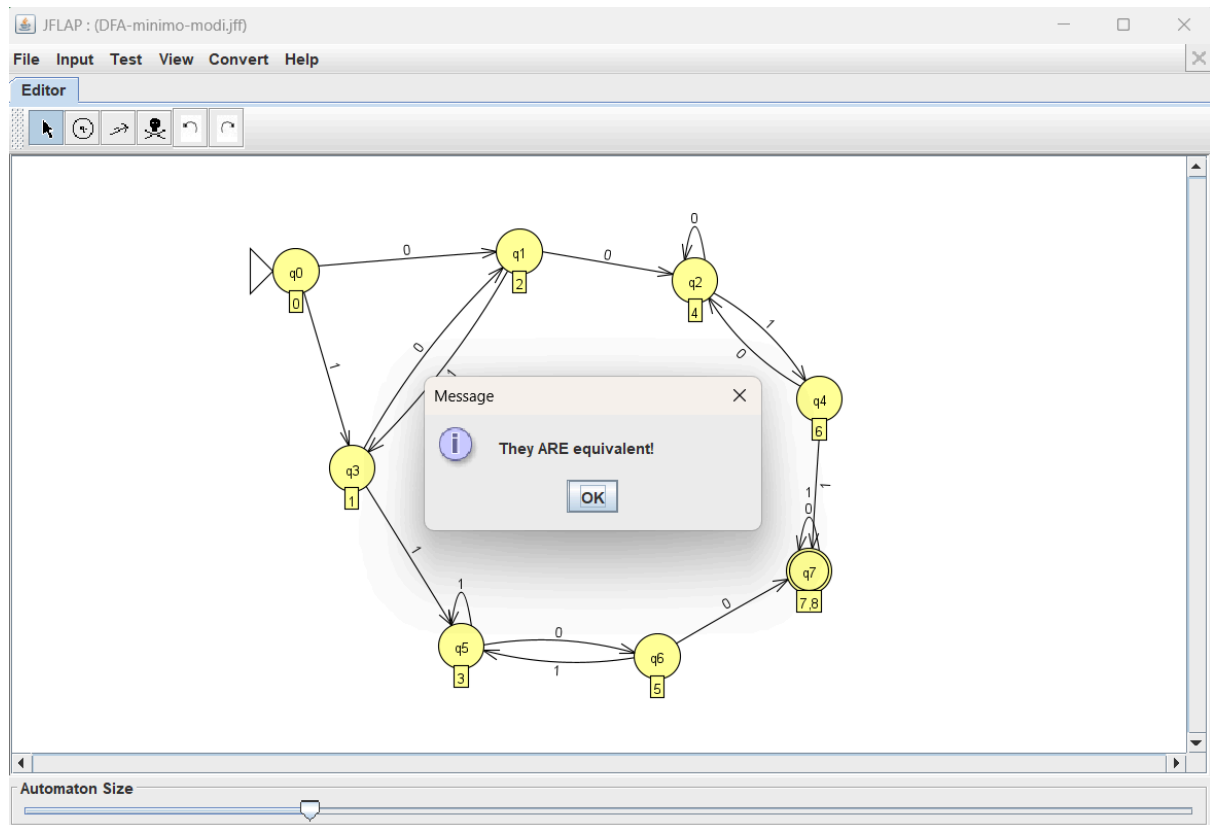


Figura 53: El DFA mínimo obtenido es equivalente al NFA del que partimos.