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

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

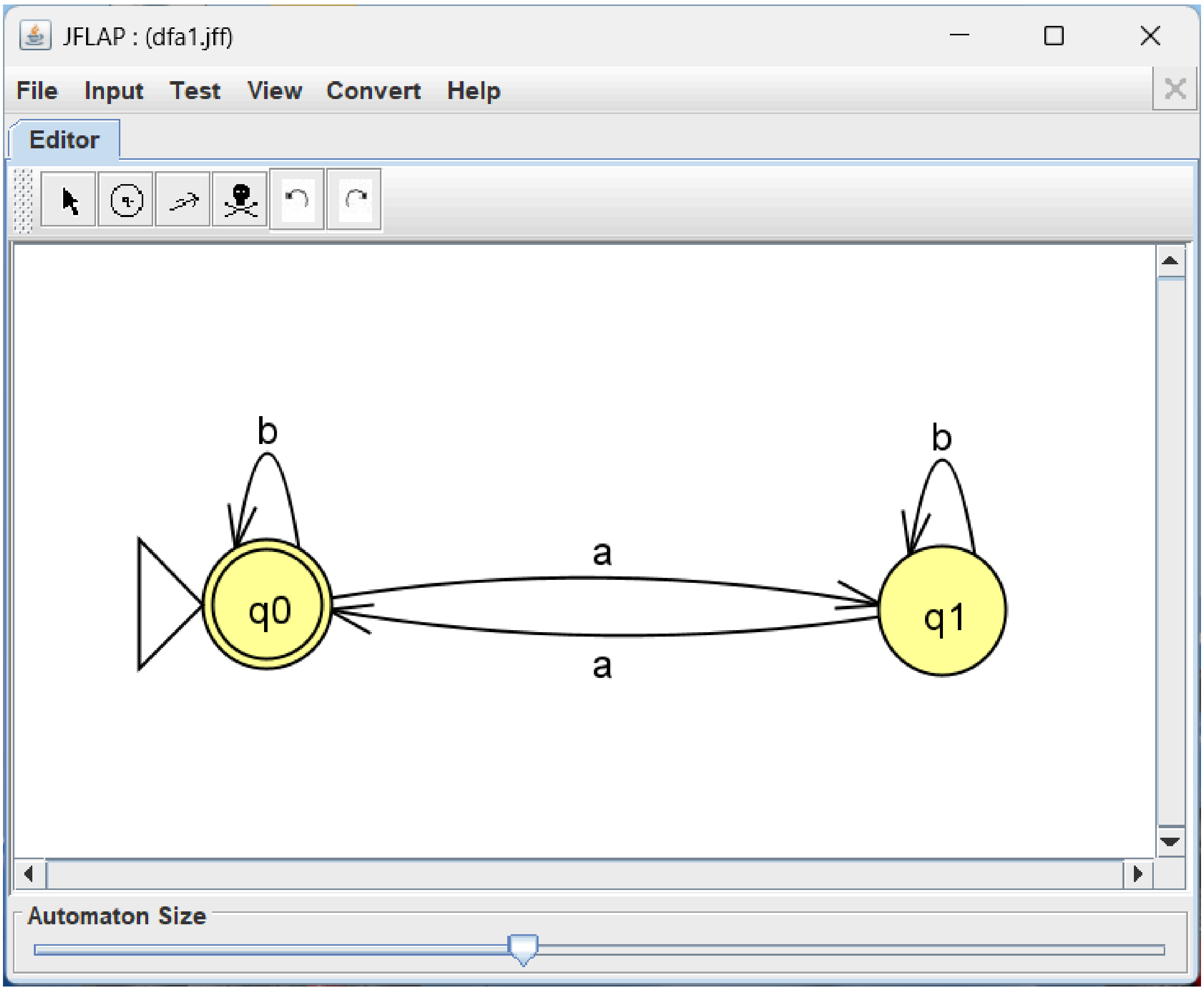

Figura 1: DFA que reconoce cadenas con número de “a’s” par sobre el alfabeto Σ = {a, b}

El estado q0 representa el número par de “a’s” y la cantidad de “b’s” que sea, mientras que el estado q1 representa una cantidad impar de “a’s” y la cantidad de “b’s” que sea

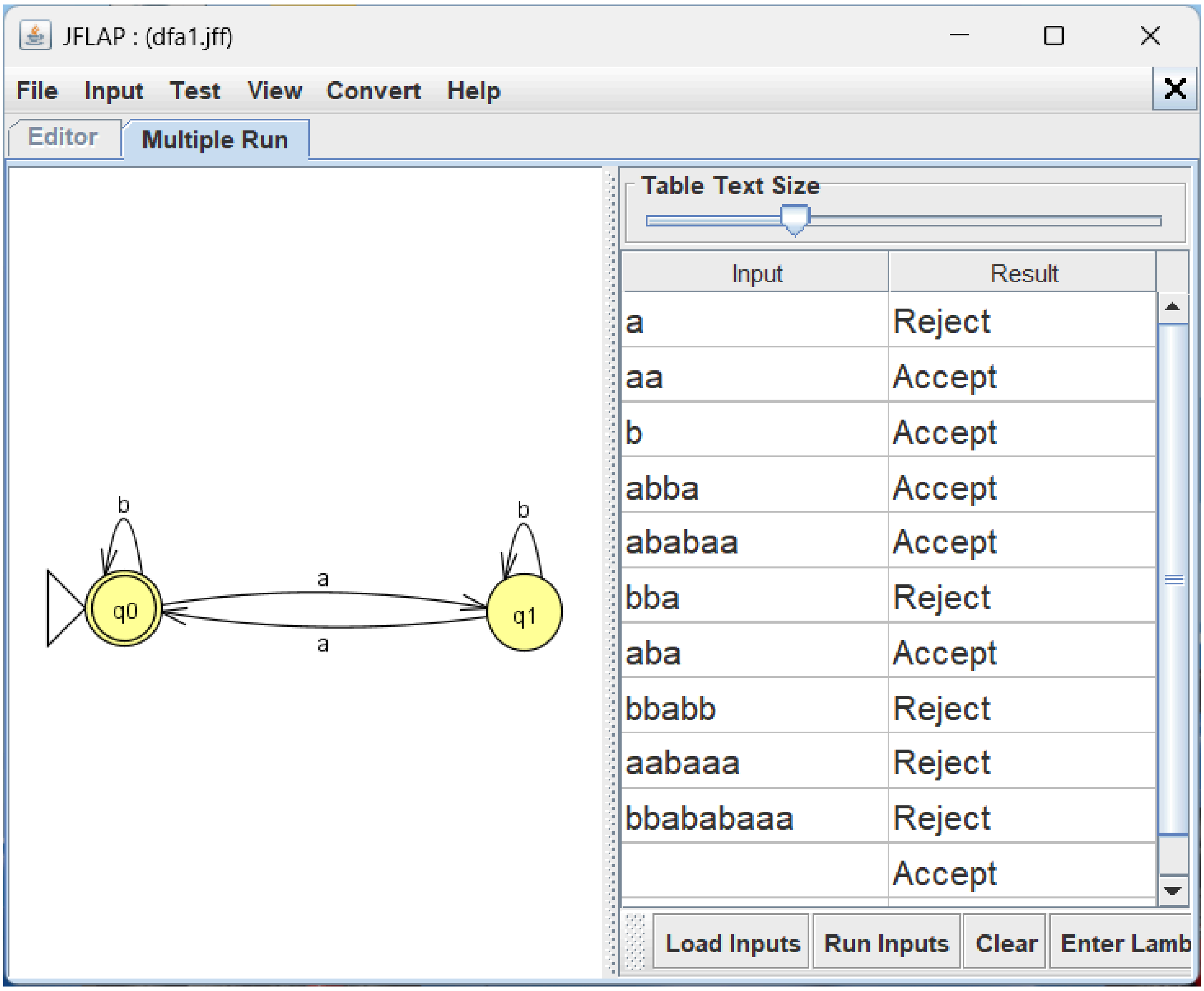

Figura 2: Cadenas de prueba para el DFA anterior

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

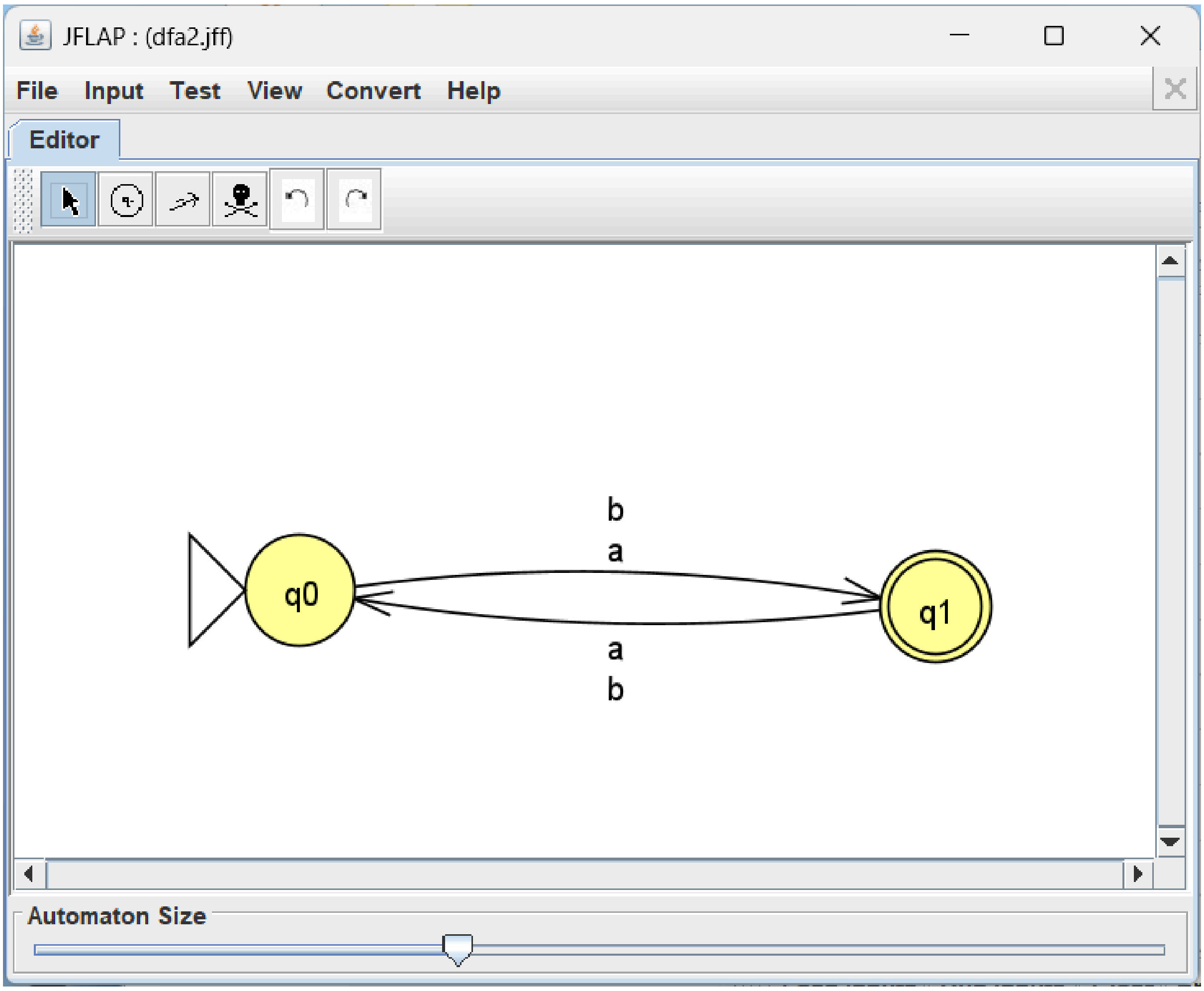

Figura 3: DFA que reconoce cadenas de longitud impar sobre el alfabeto Σ = {a, b}

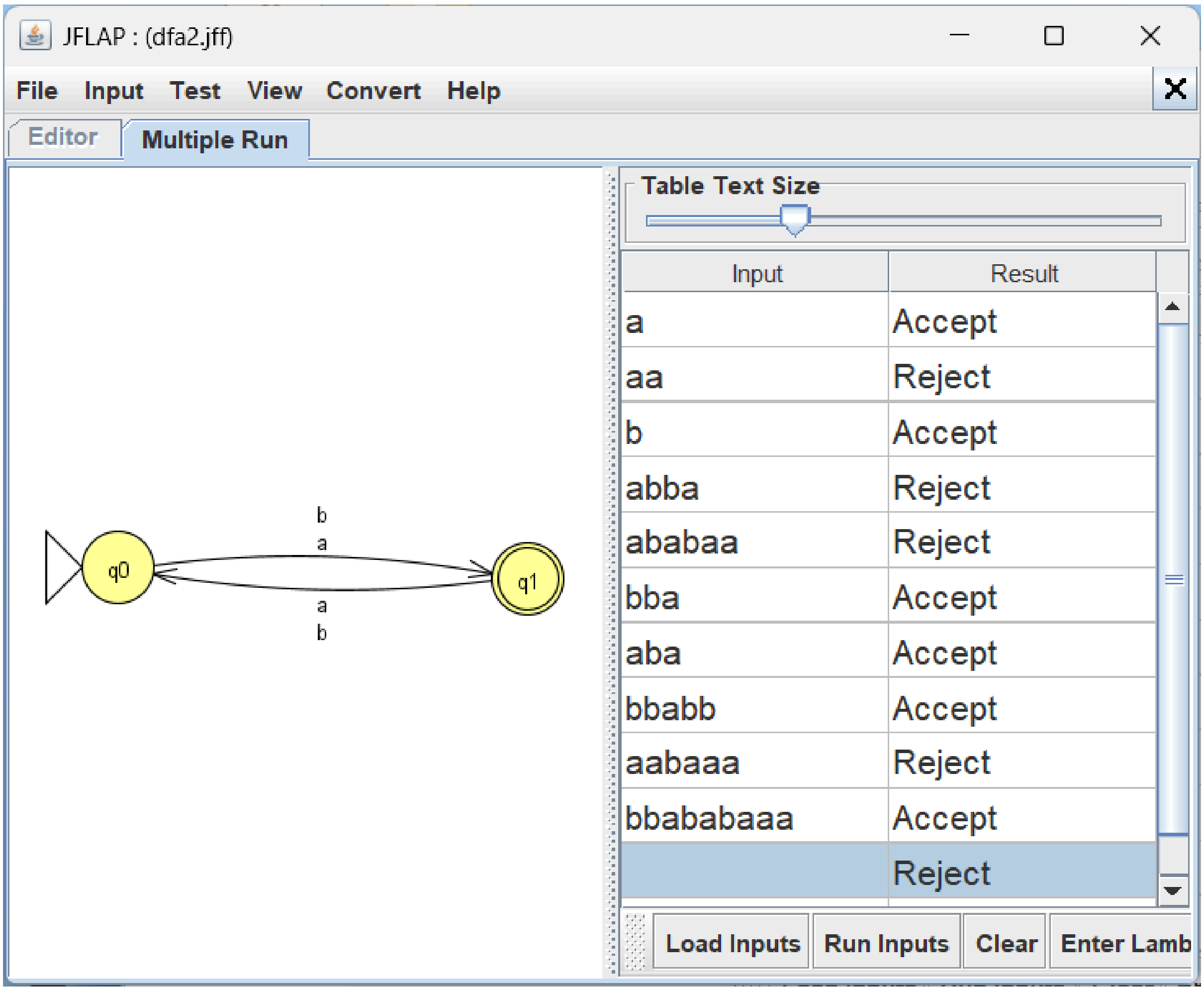

Figura 4: Cadenas de prueba para el DFA anterior

1.3. Diseñar un autómata finito determinista que reconozca cadenas sobre el alfabeto Σ = {a, b} con número de "a's" par o longitud impar.

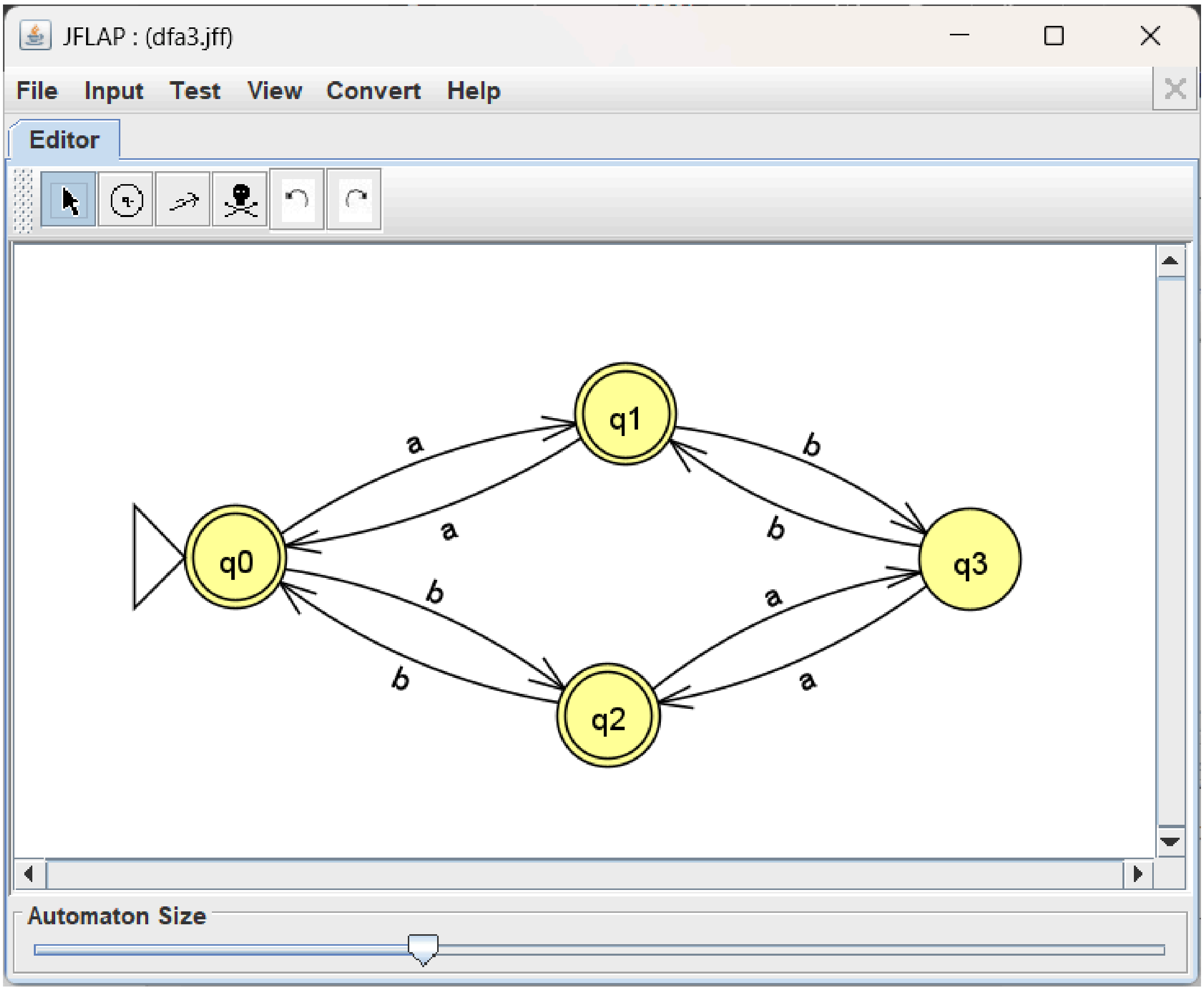

Figura 5: DFA que reconoce cadenas con número de "a's" par o longitud impar sobre el alfabeto Σ = {a, b}

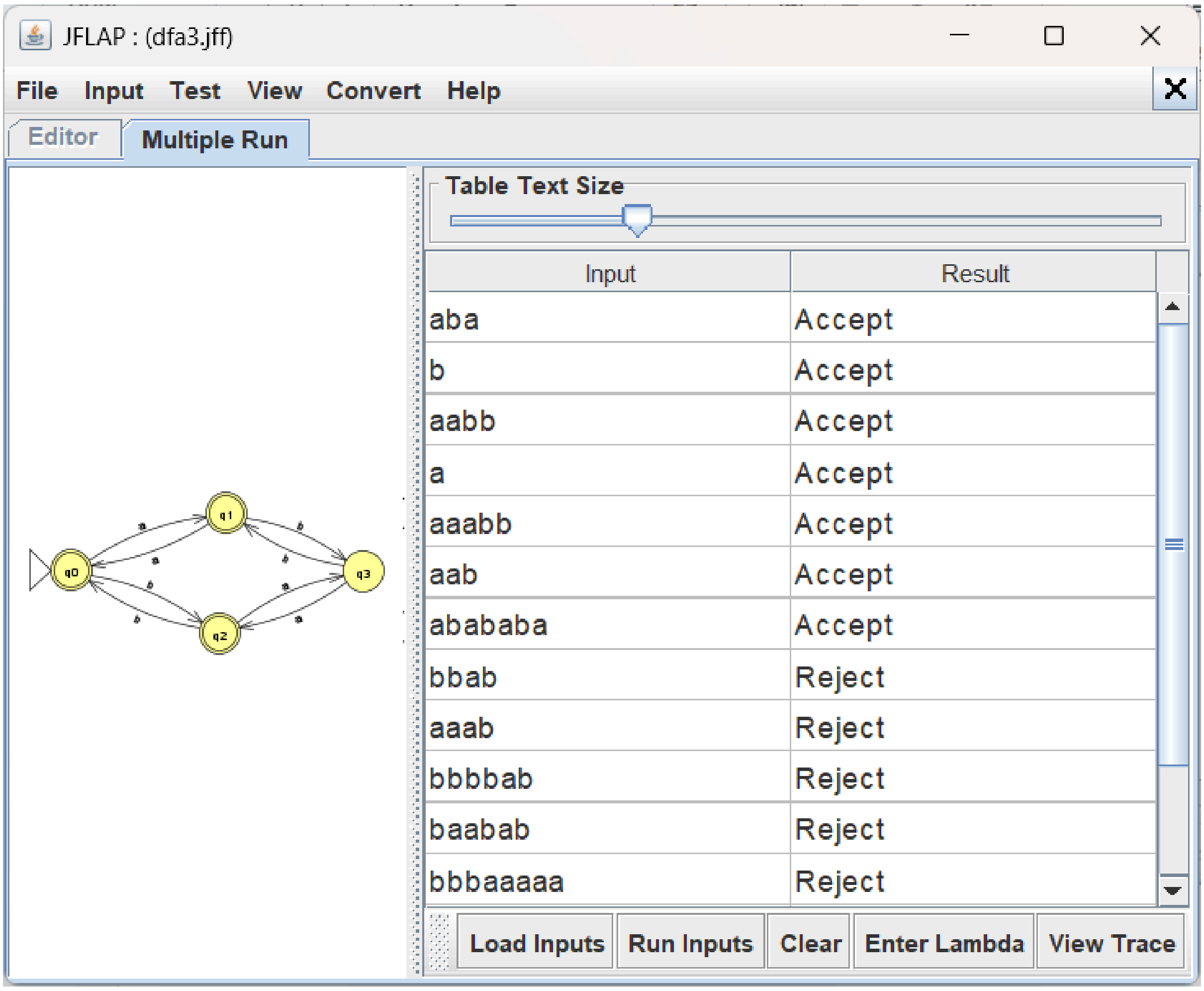

Figura 6: Cadenas de prueba para el DFA anterior

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

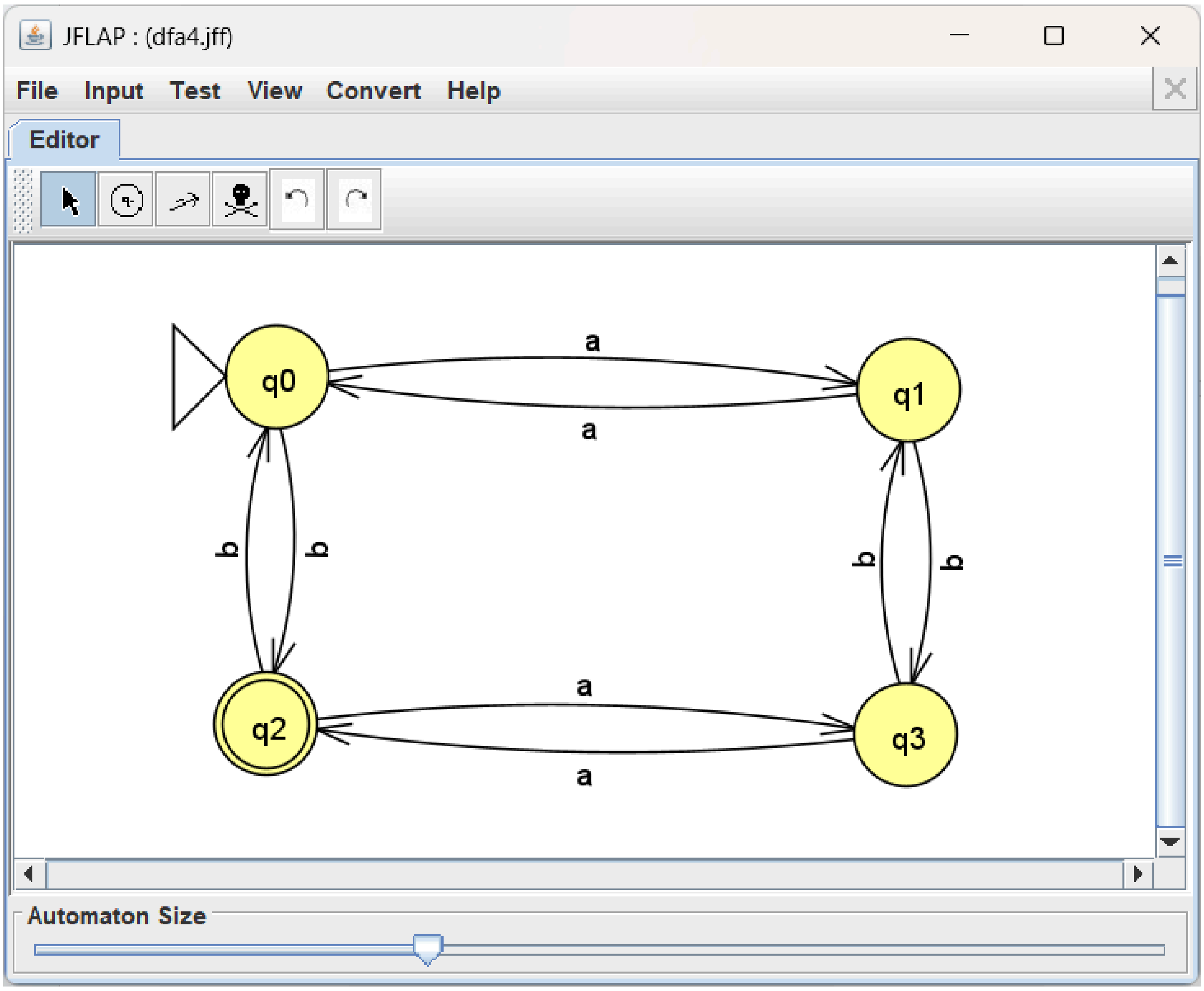

Figura 7: DFA que reconoce cadenas con número de “a’s” par y longitud impar sobre el alfabeto Σ = {a, b}

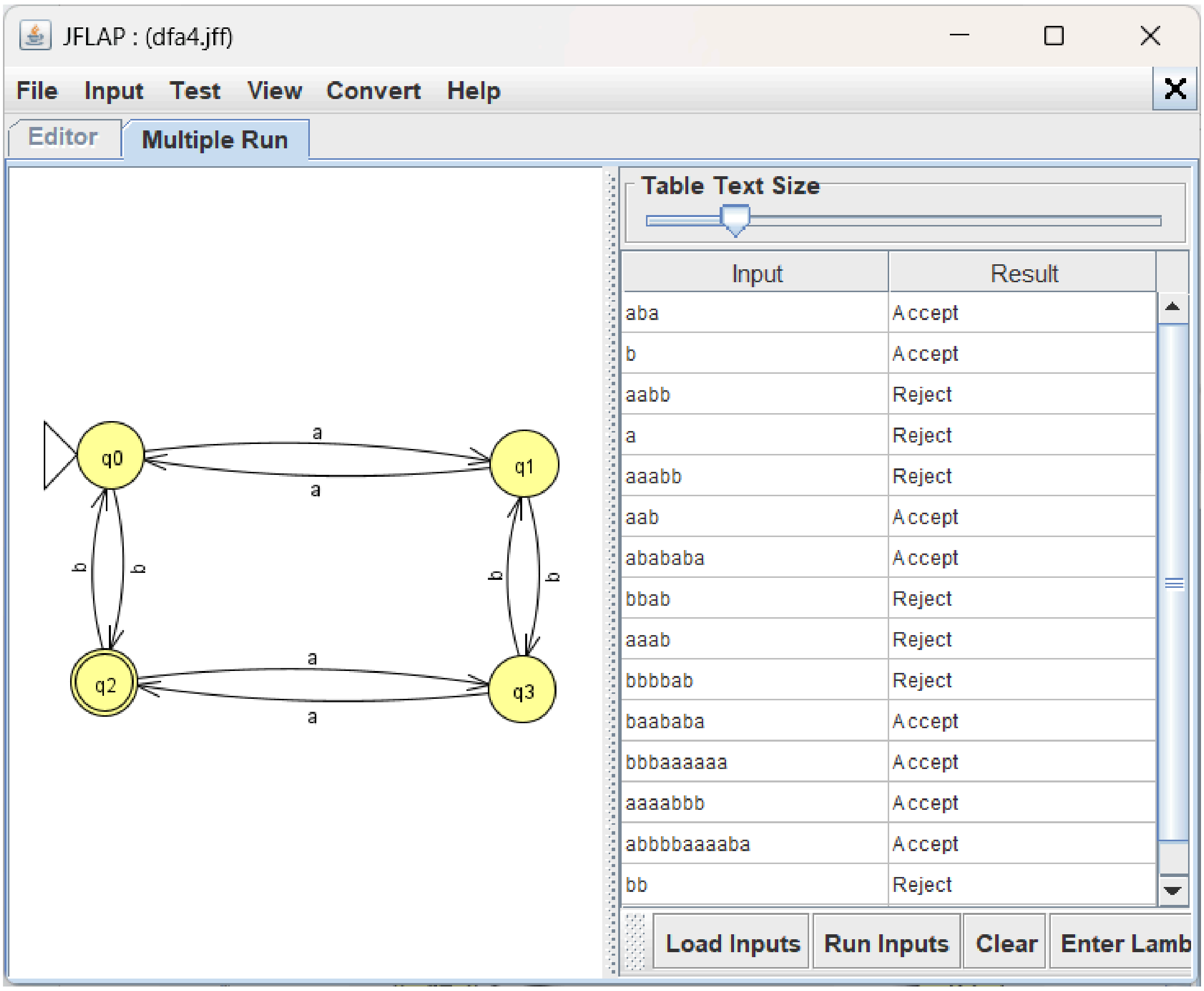

Figura 8: Cadenas de prueba para el DFA anterior

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

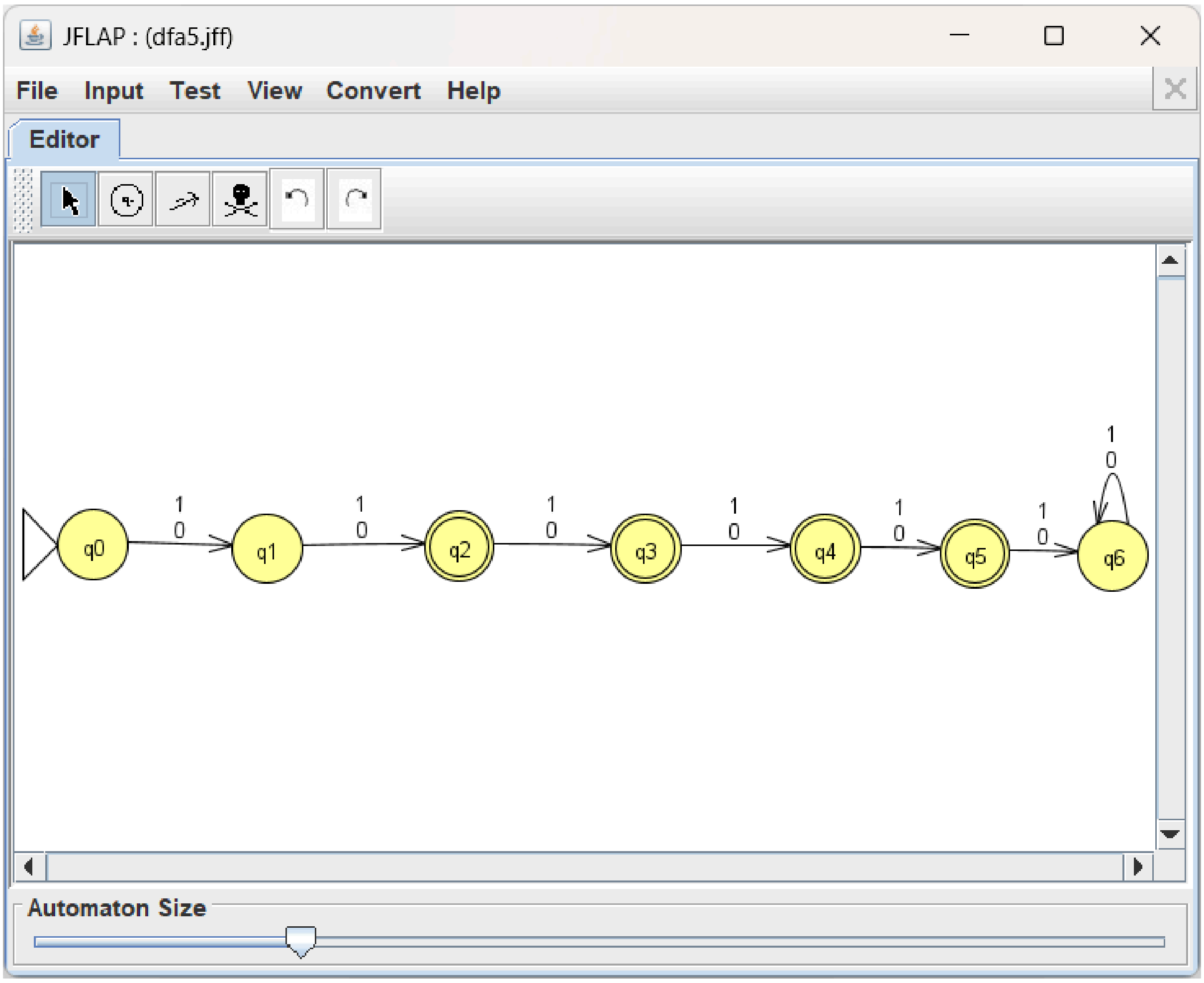

Figura 9: DFA que reconoce cadenas w sobre el alfabeto {0, 1} tales que 2 ≤ |w| ≤ 5.

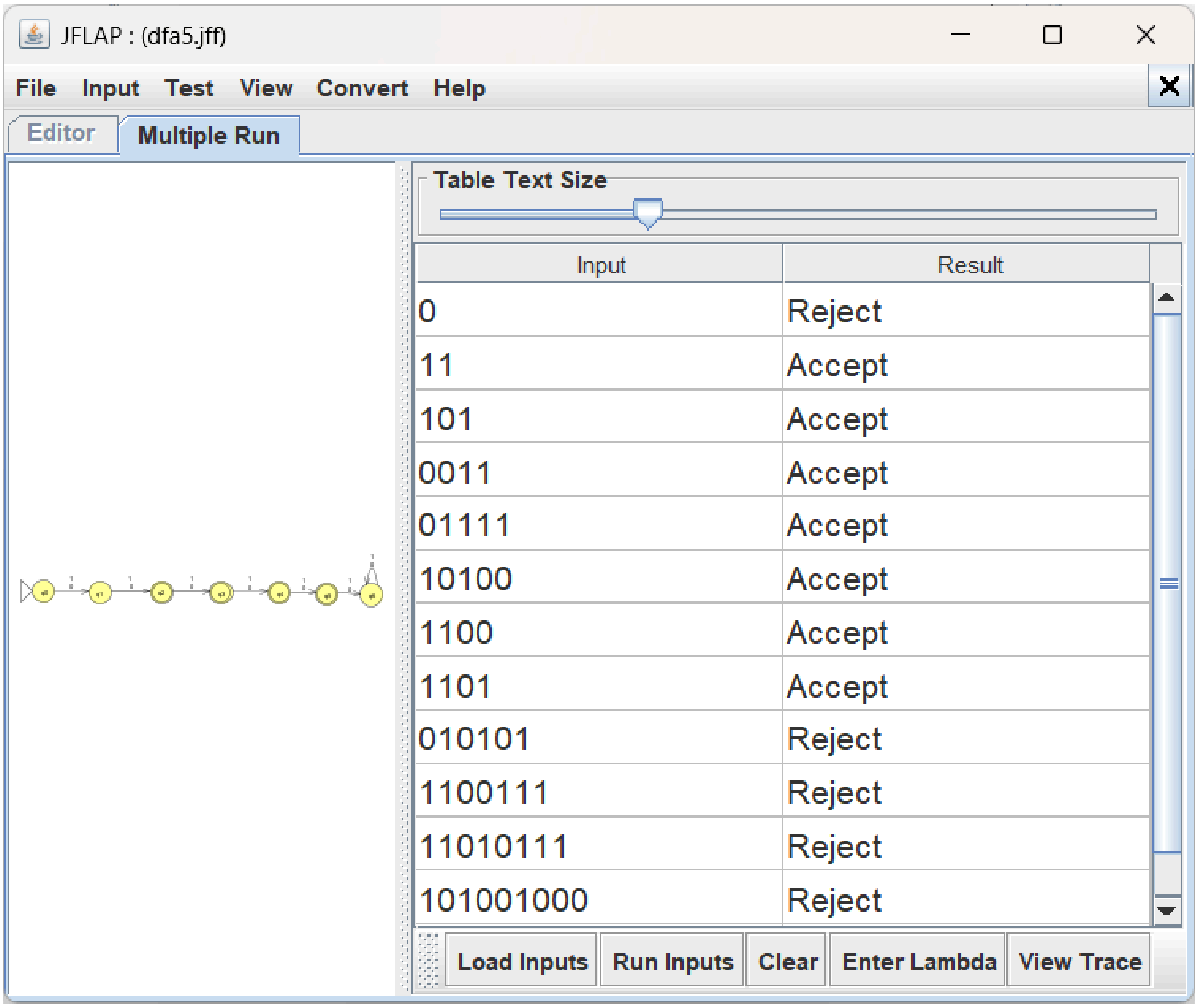

Figura 10: Cadenas de prueba para el DFA anterior.

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

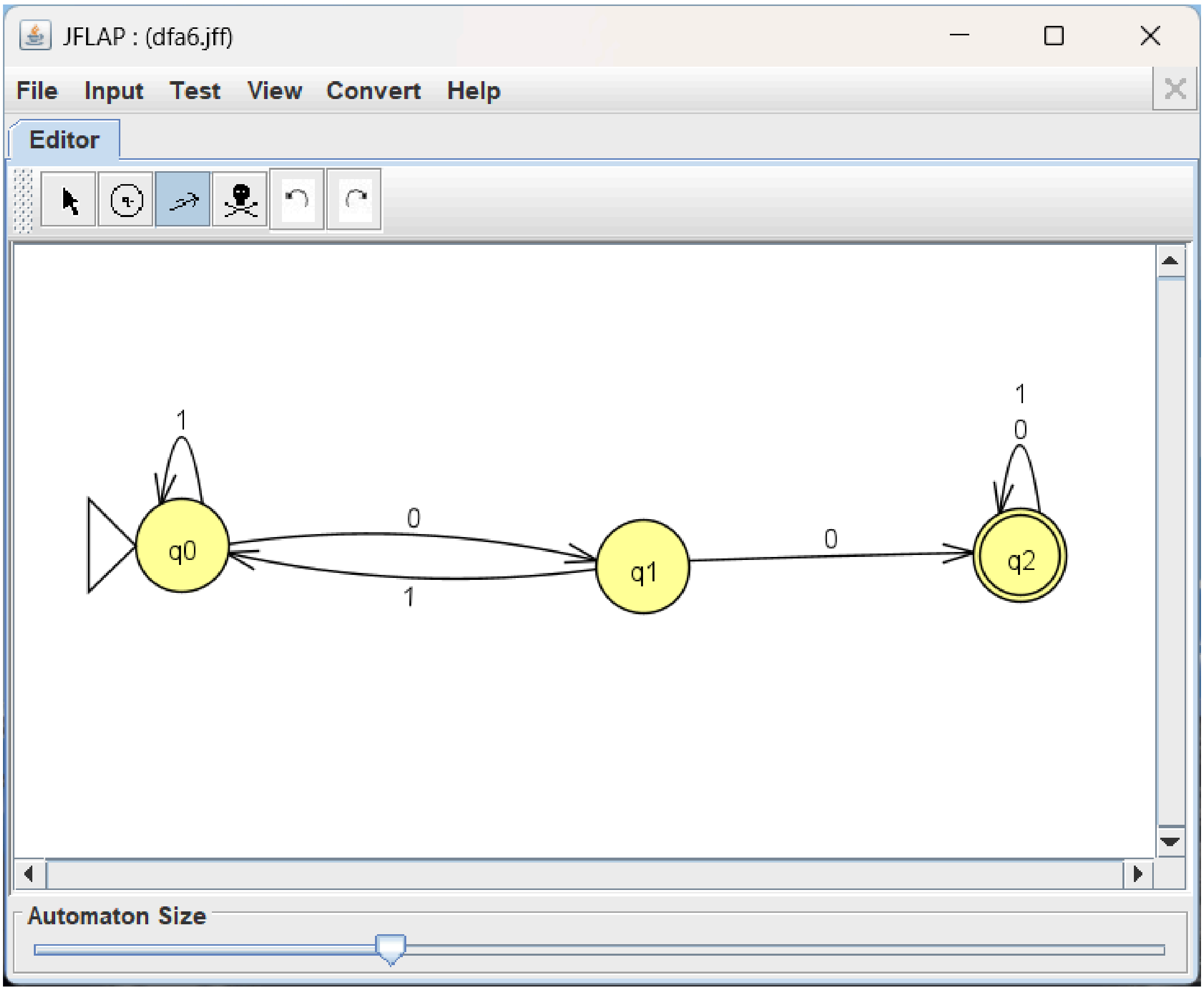

Figura 11: DFA que reconoce cadenas que tengan como mínimo dos ceros consecutivos sobre el alfabeto {0, 1} .

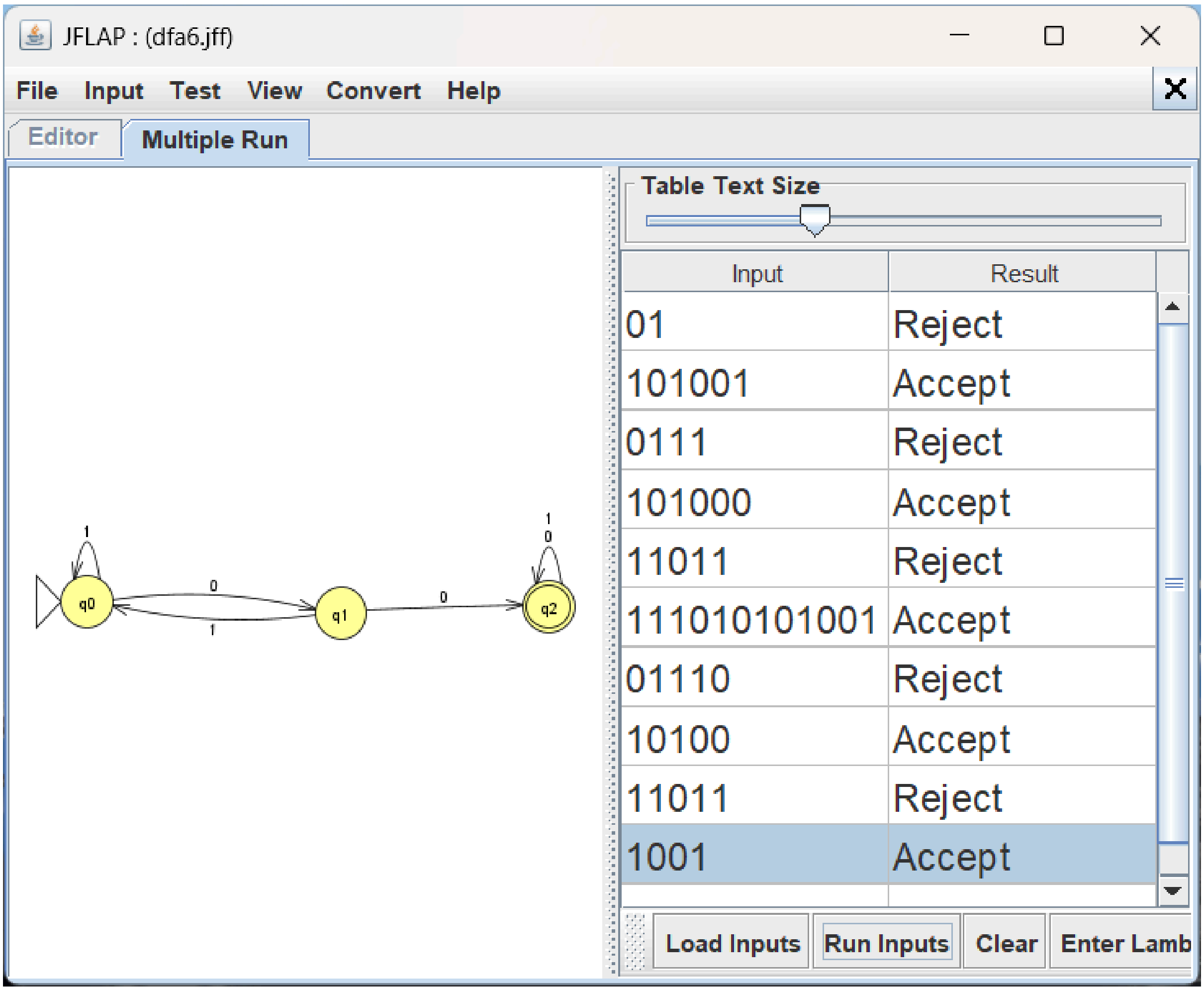

Figura 12: Cadenas de prueba para el DFA anterior.

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

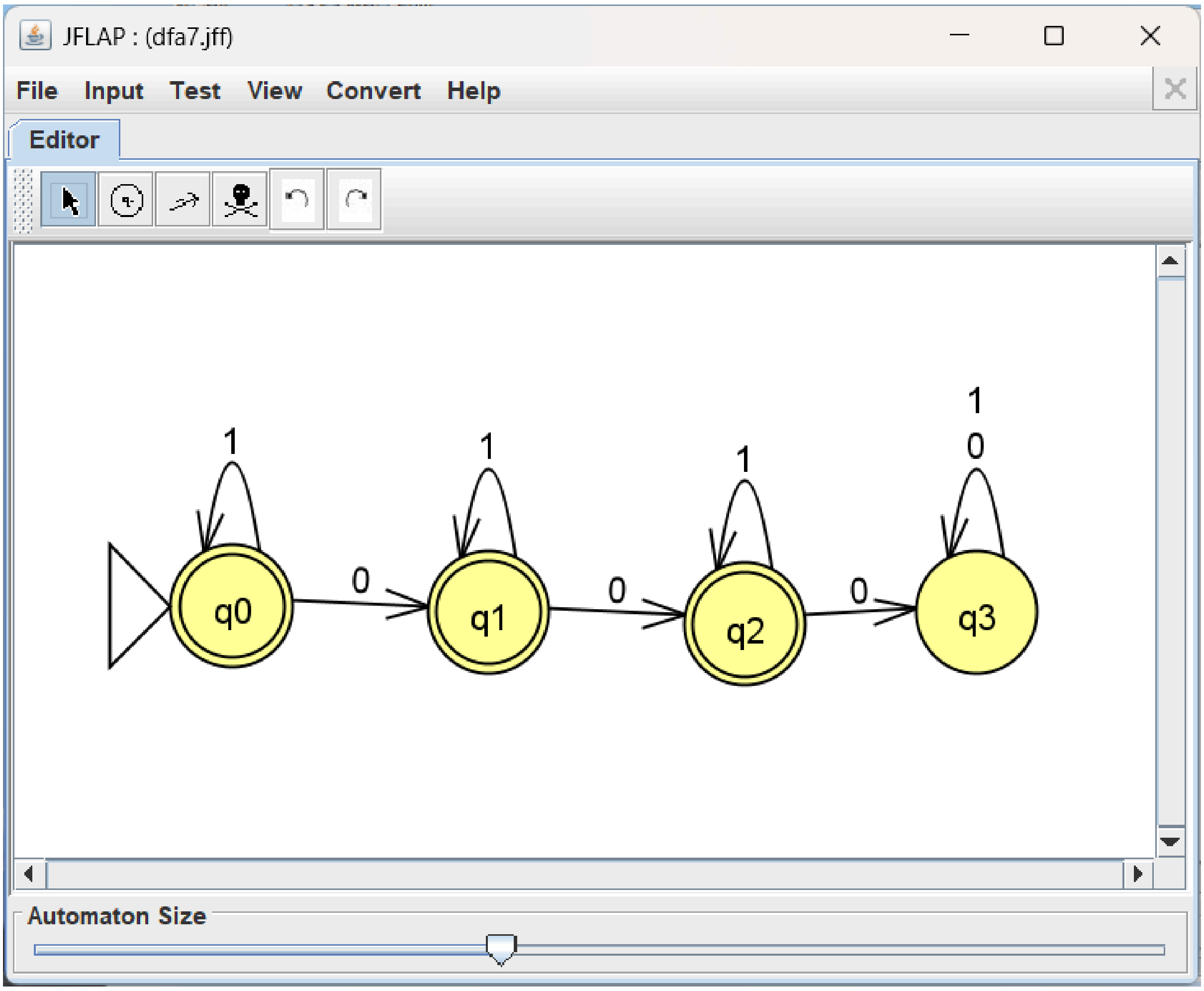

Figura 13: DFA que reconoce cadenas que tengan como máximo dos ceros sobre el alfabeto {0, 1}

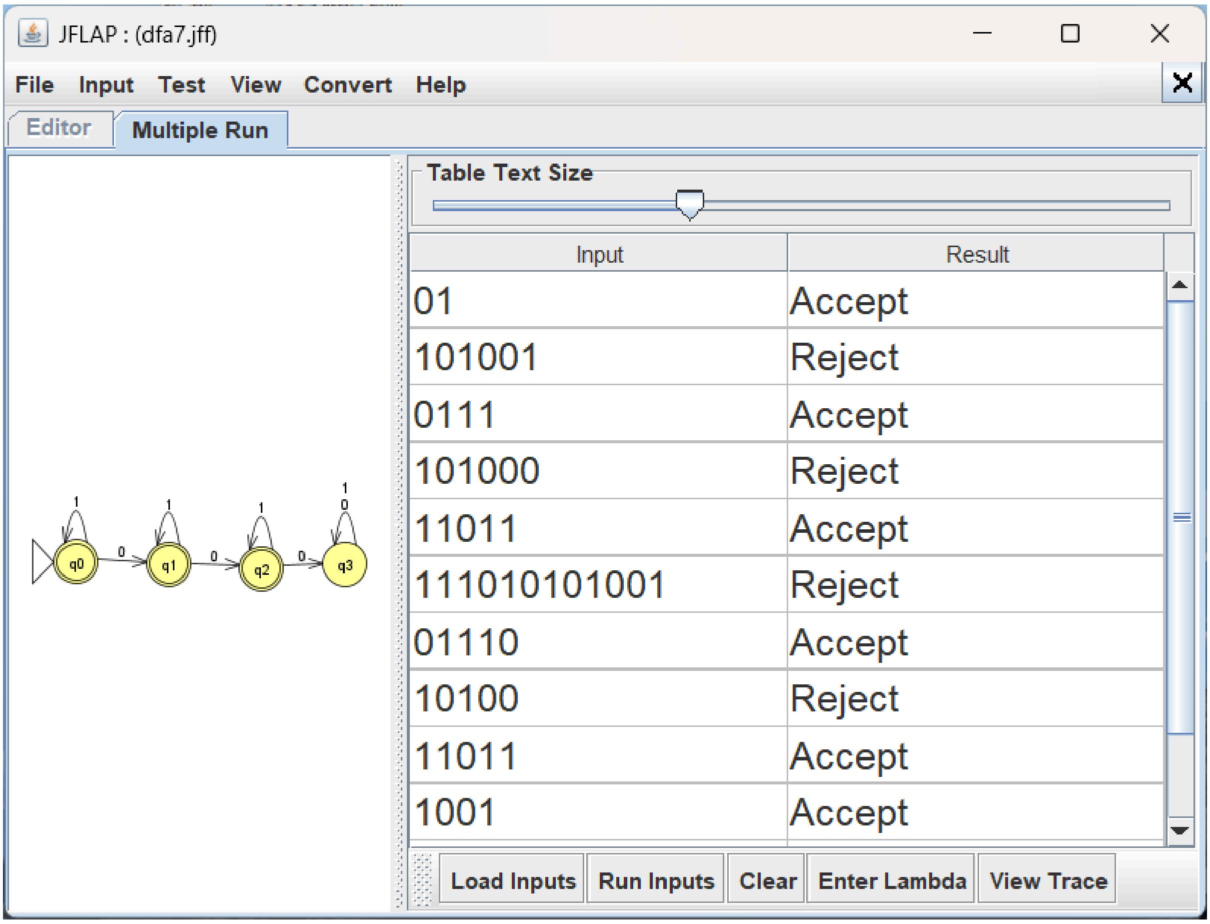

Figura 14: Cadenas de prueba para el DFA anterior.

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

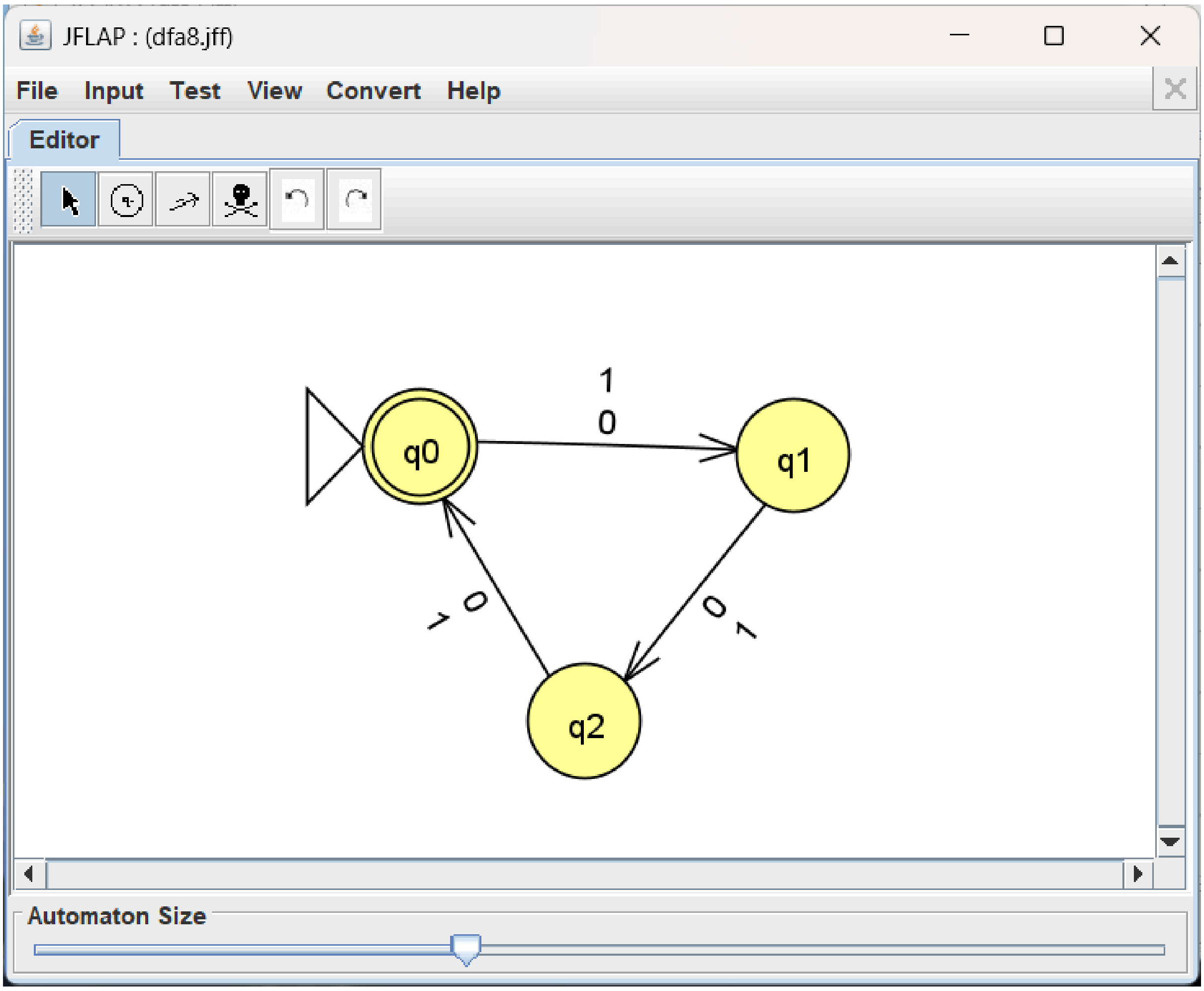

Figura 15: DFA que reconoce cadenas con longitud múltiplo de 3 sobre el alfabeto {0, 1}

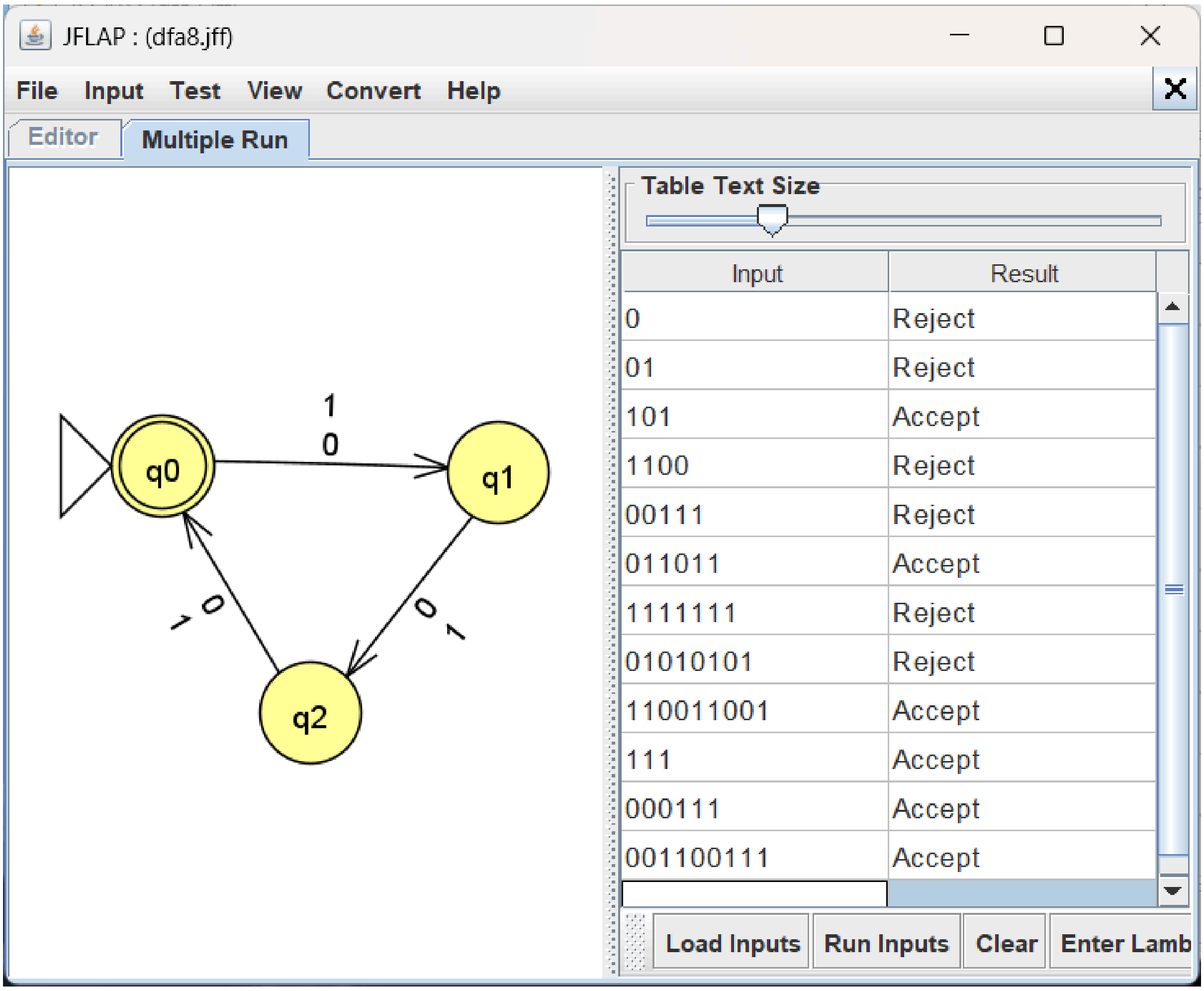

Figura 16: Cadenas de prueba para el DFA anterior.

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

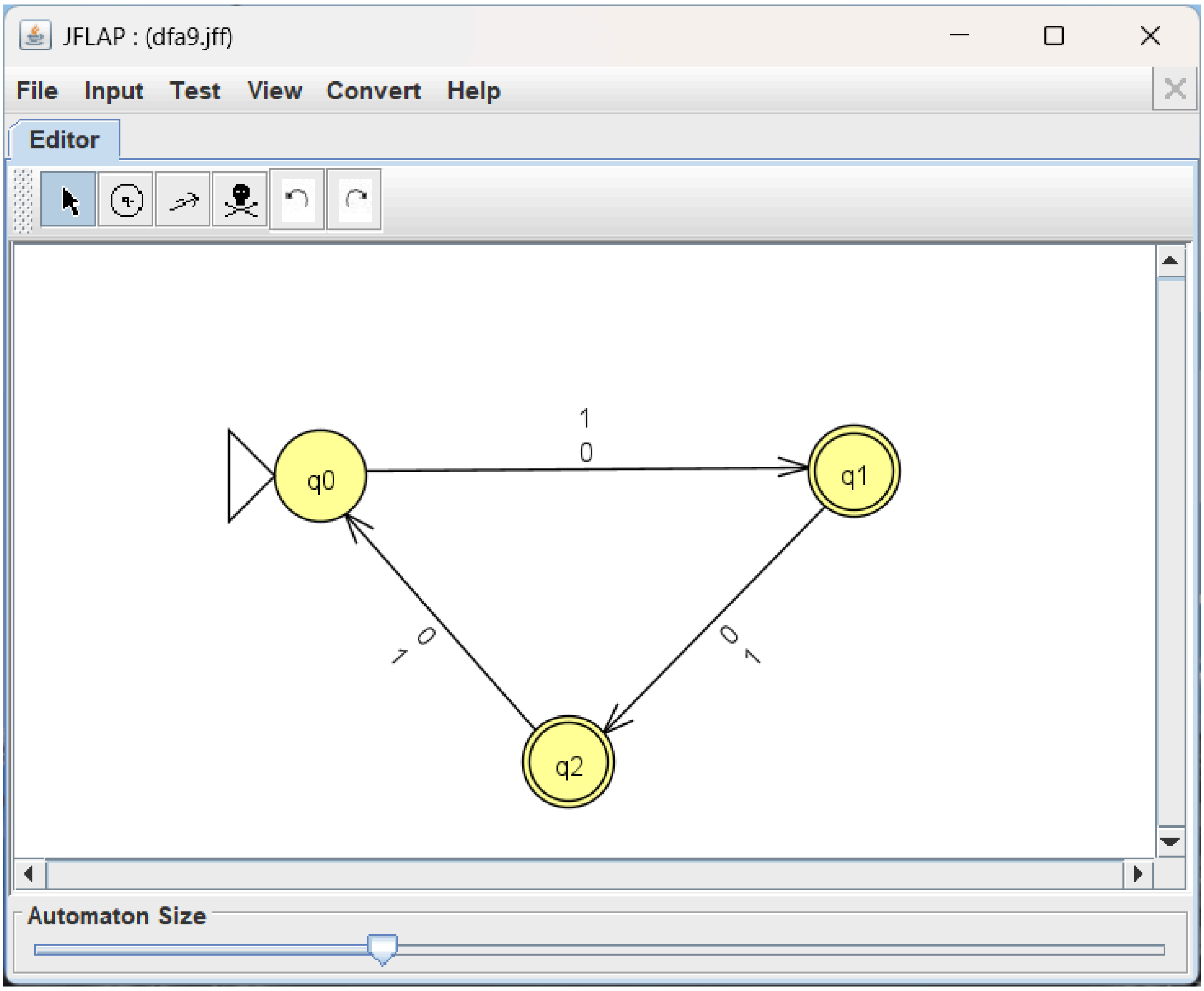

Figura 17: DFA que reconoce cadenas con longitud que no sea múltiplo de 3 sobre el alfabeto {0, 1}

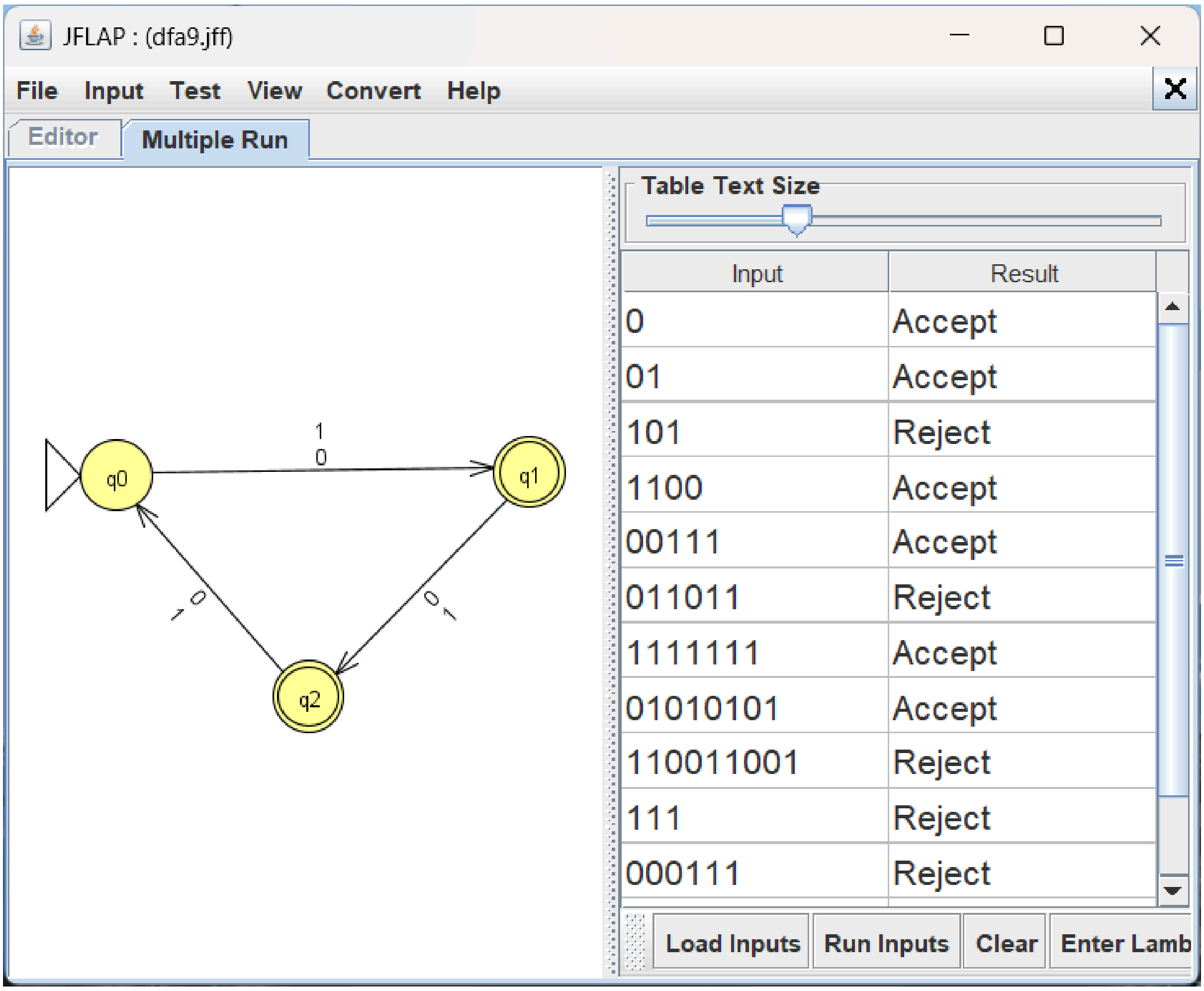

Figura 18: Cadenas de prueba para el DFA anterior.

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

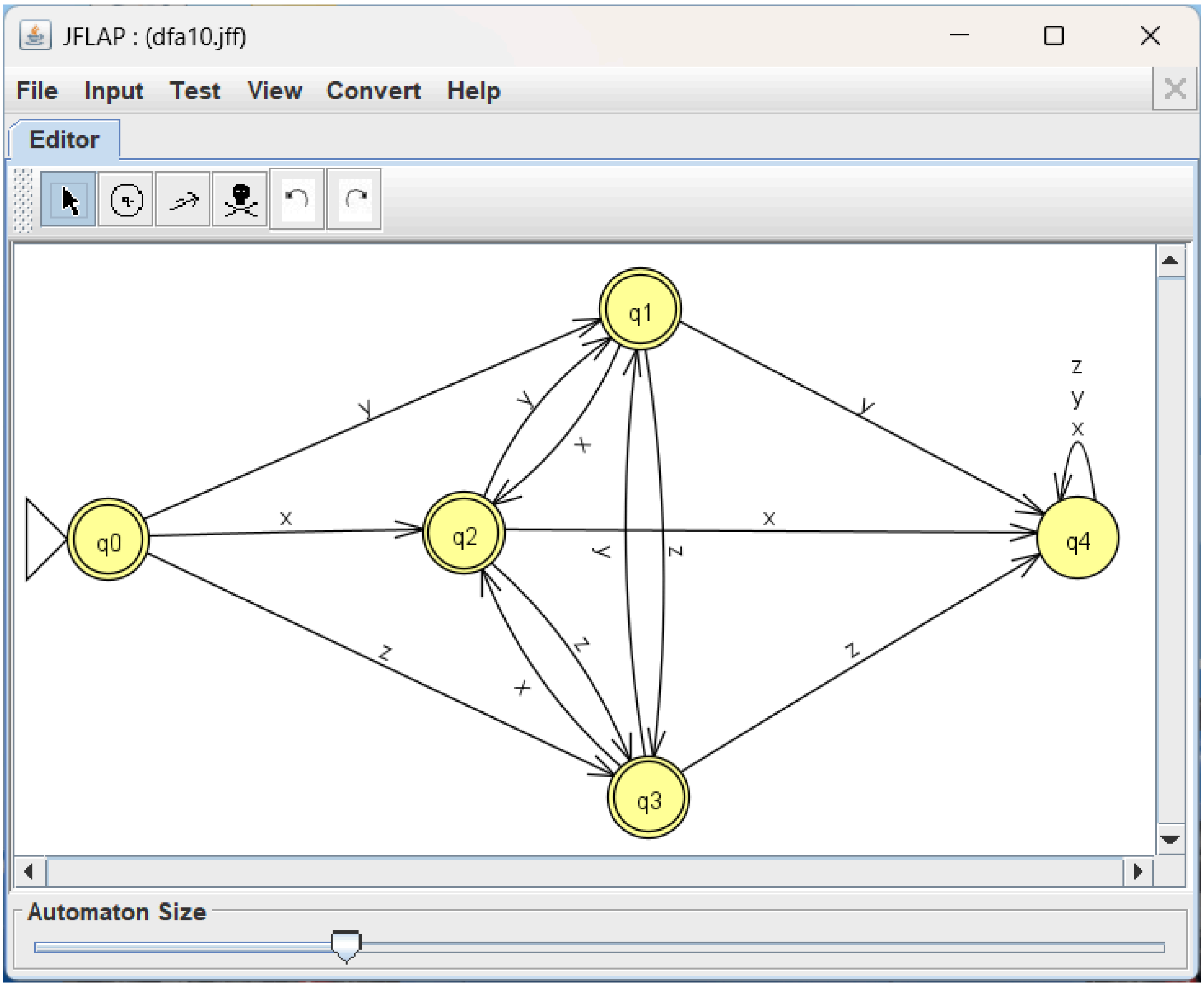

Figura 19: DFA que reconoce cadenas que no contengan dos símbolos iguales consecutivos sobre el alfabeto {x, y, z}

.

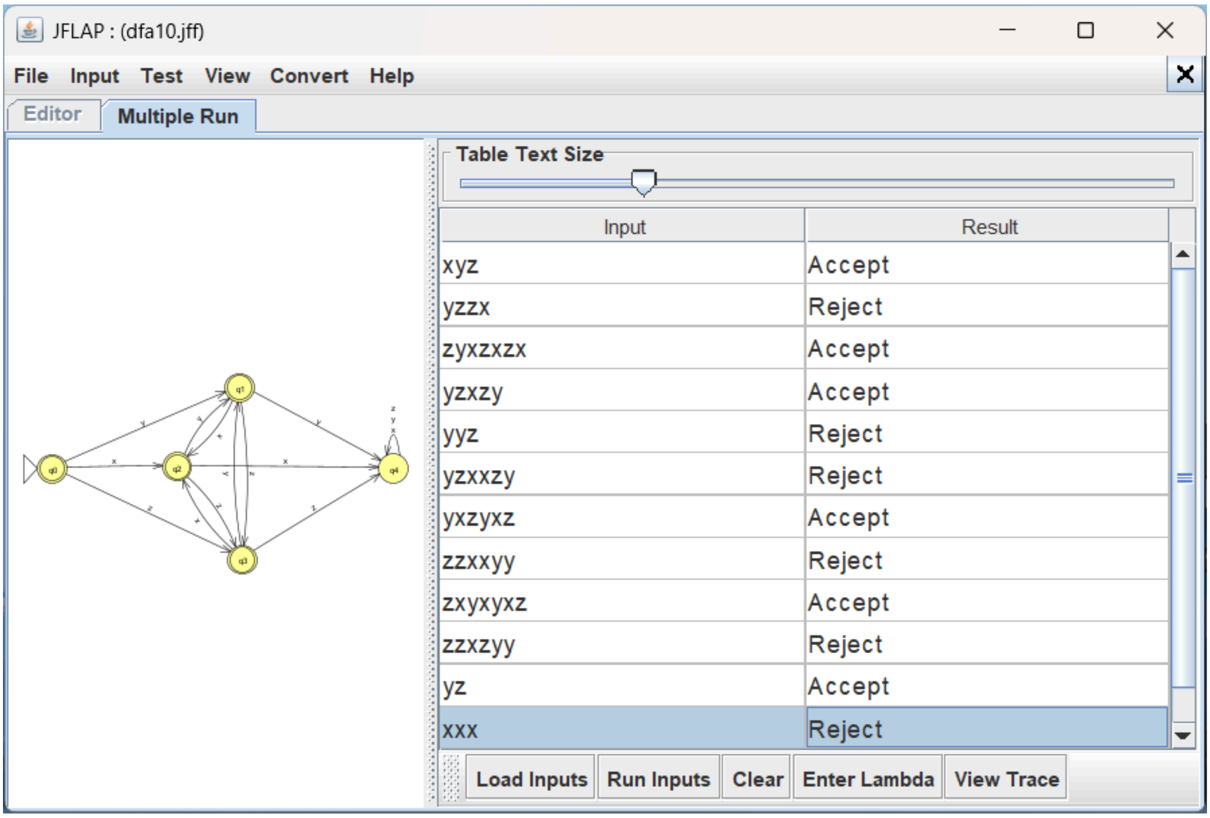

Figura 20: Cadenas de prueba para el DFA anterior.

2. Diseño de NFAs

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

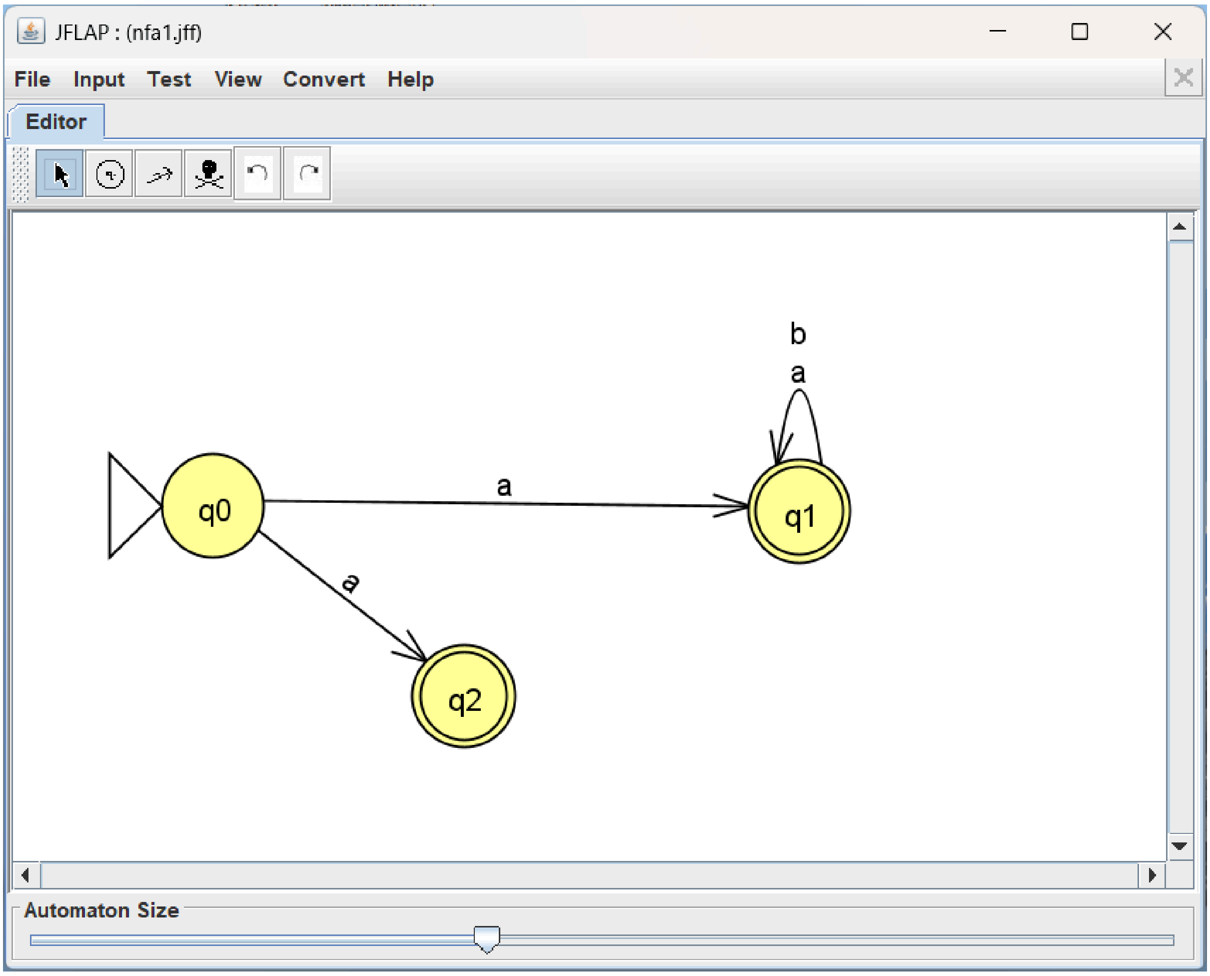

Figura 21: NFA que reconoce cadenas que empiecen por "a" sobre el alfabeto Σ = {a, b}

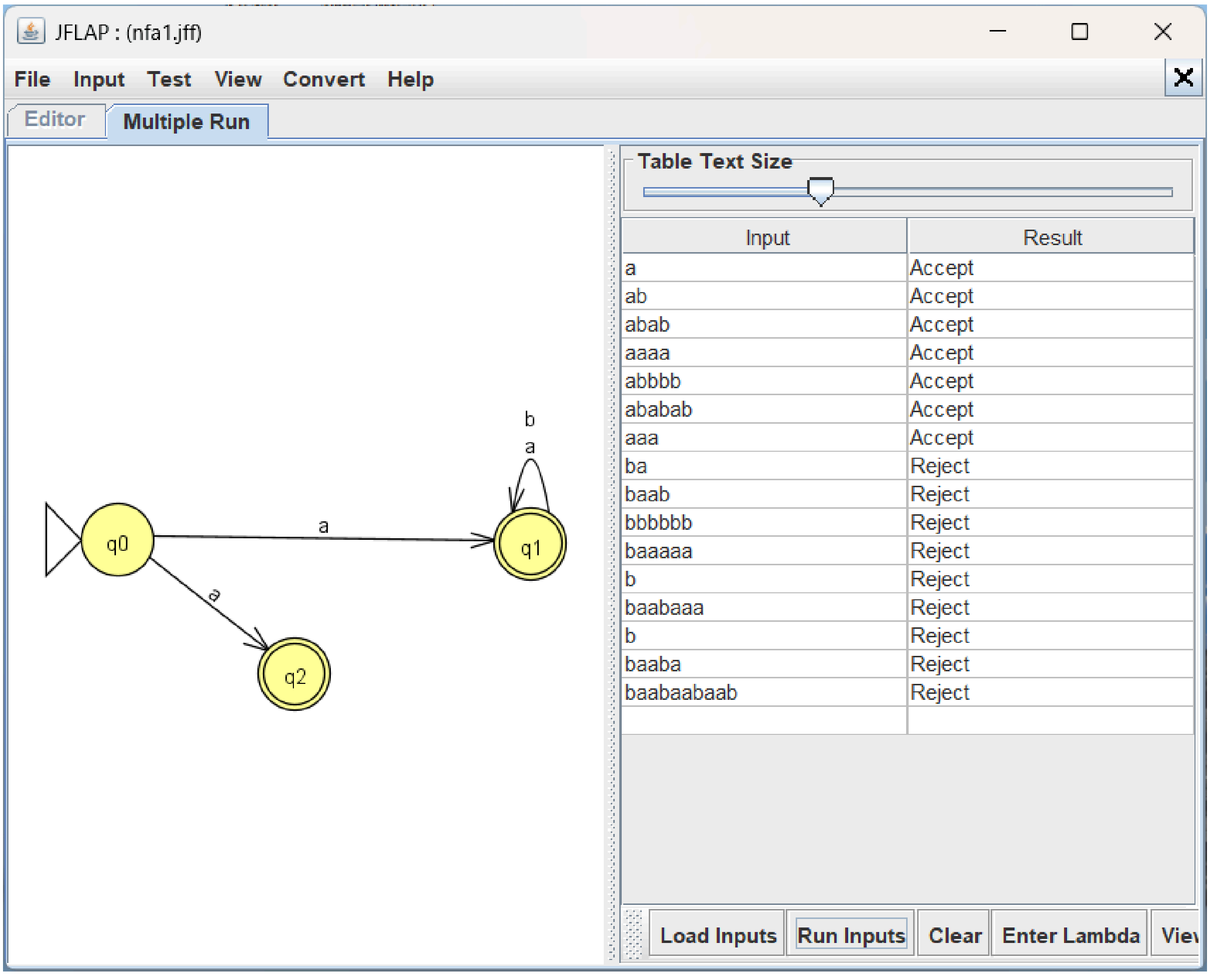

Figura 22: Cadenas de prueba para el NFA anterior.

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

S = {q0} = A
δ(A, a) = {q1, q2} = B
δ(A, b) = ∅
δ(B, a) = {q1} = C
δ(B, b) = {q1} = C
δ(C, a) = {q1} = C
δ(C, b) = {q1} = C

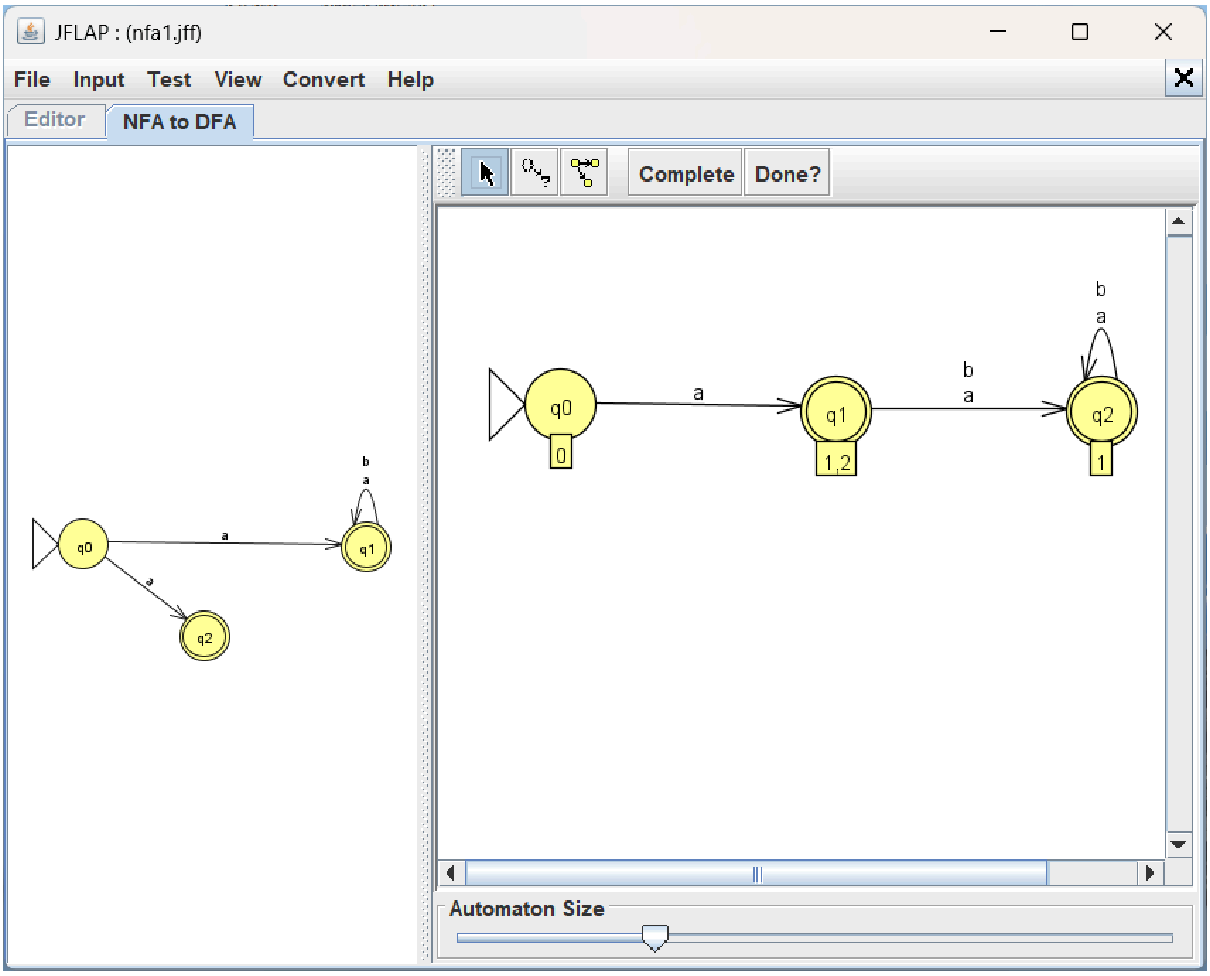

Figura 23: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{B, C}, {A, ∅}}
π’ = {{B,C}, {A}, {∅}}

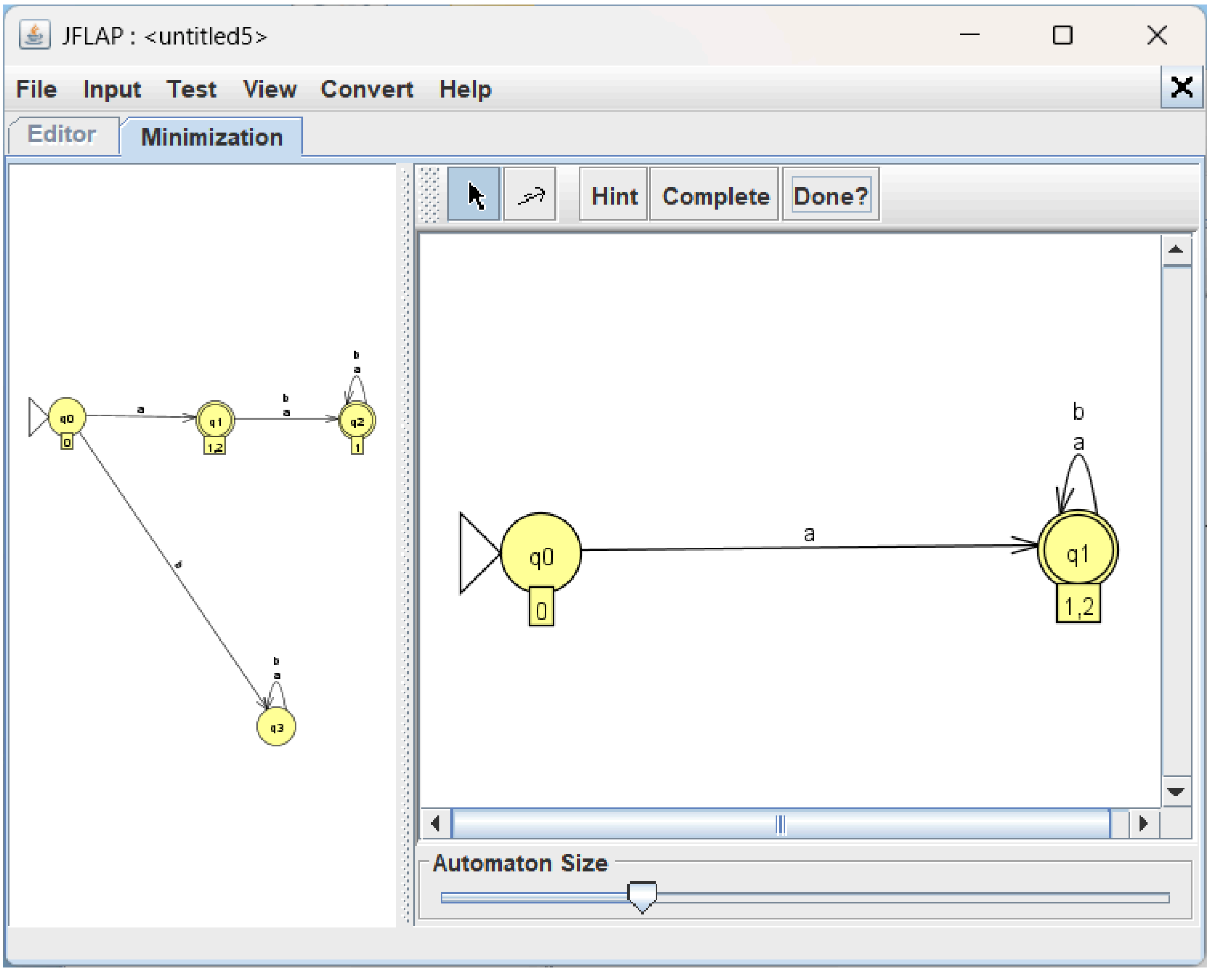

Figura 24: DFA mínimo equivalente.

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

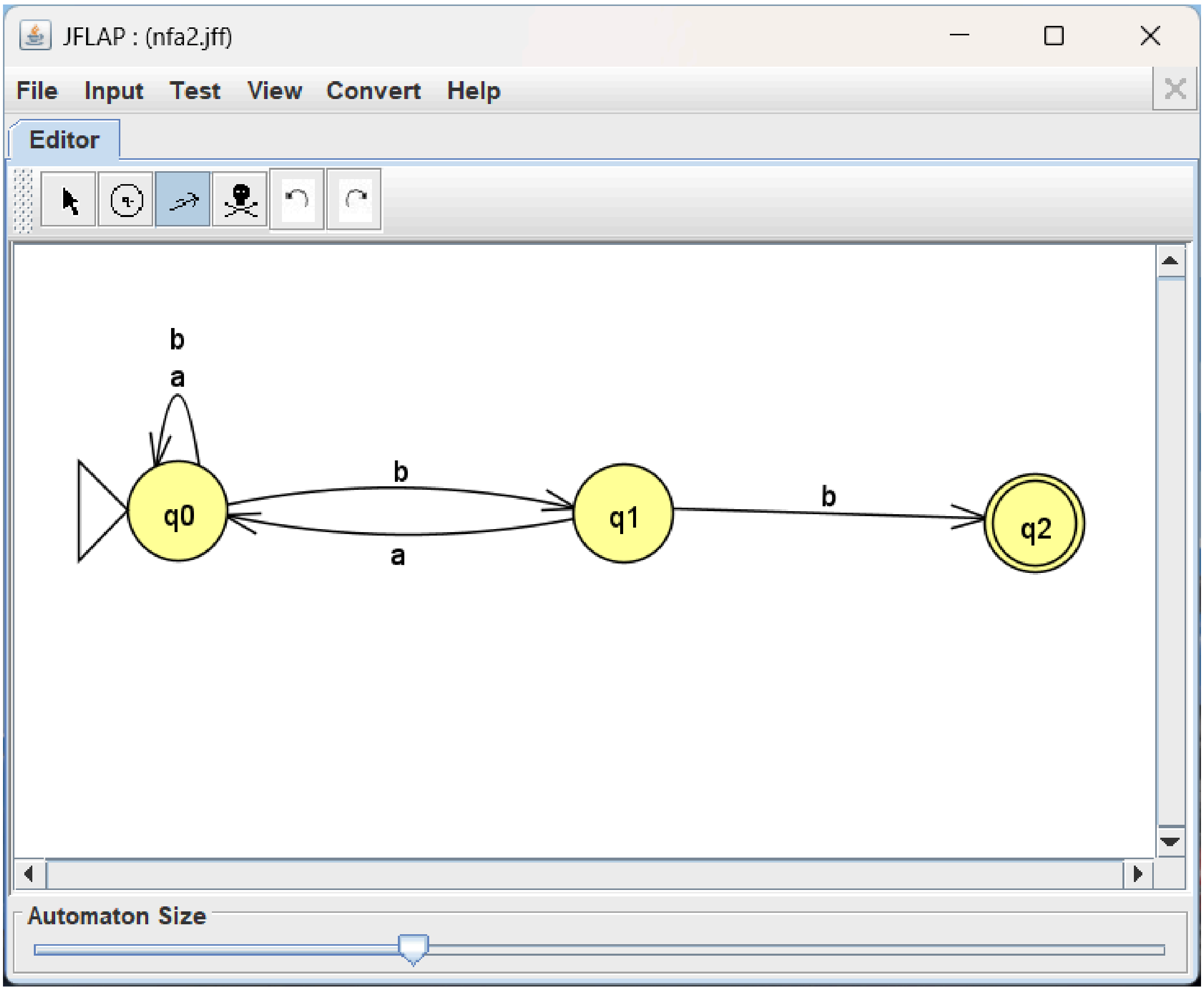

Figura 25: NFA que reconoce cadenas que que terminen en “bb” sobre el alfabeto Σ = {a, b}

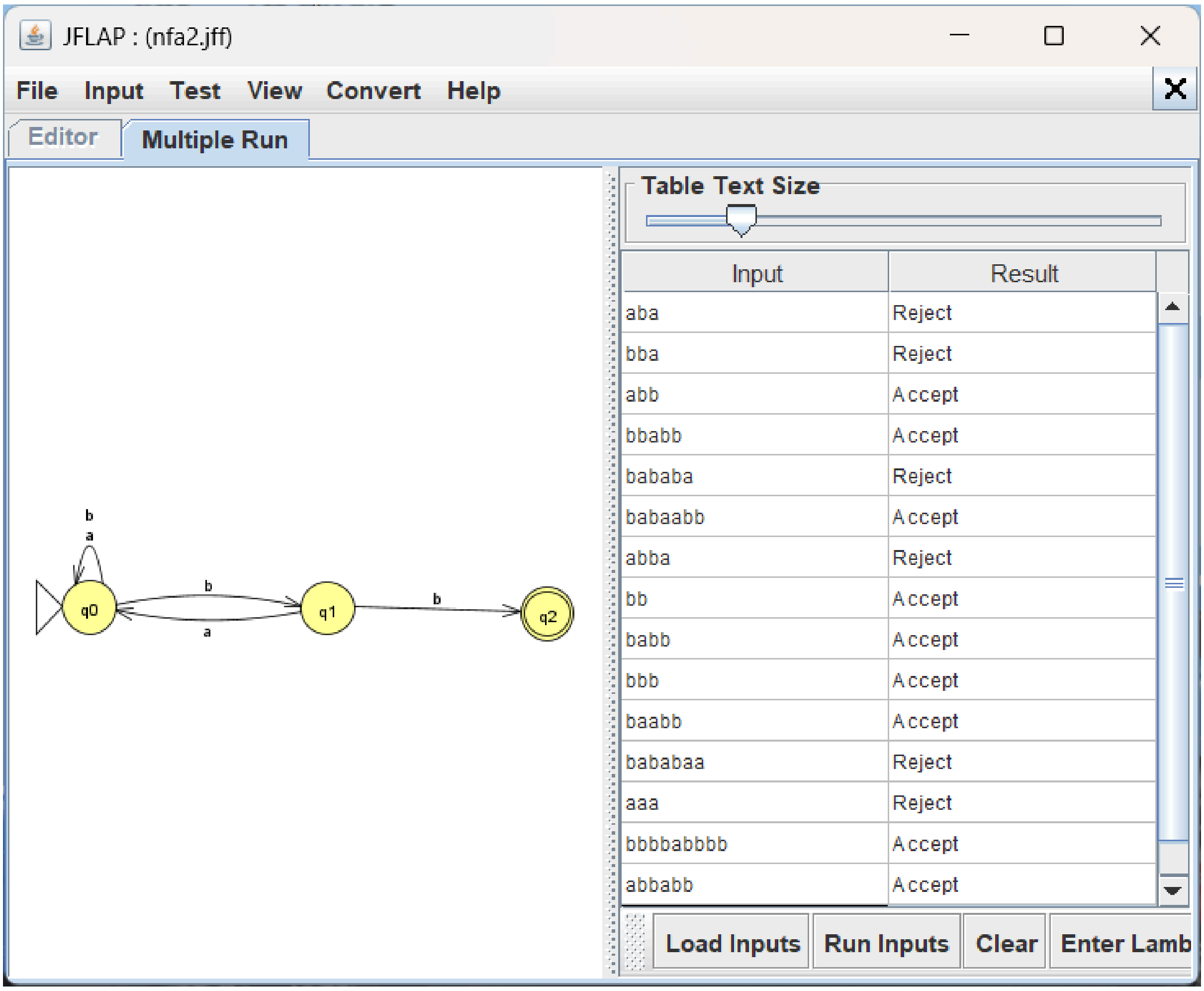

Figura 26: Cadenas de prueba para el NFA anterior.

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

S = {q0} = A
δ(A, a) = {q0} = A
δ(A, b) = {q0, q1} = B
δ(B, a) = {q0} = A
δ(B, b) = {q0, q1, q2} = C
δ(C, a) = {q0} = A
δ(C, b) = {q0, q1, q2} = C

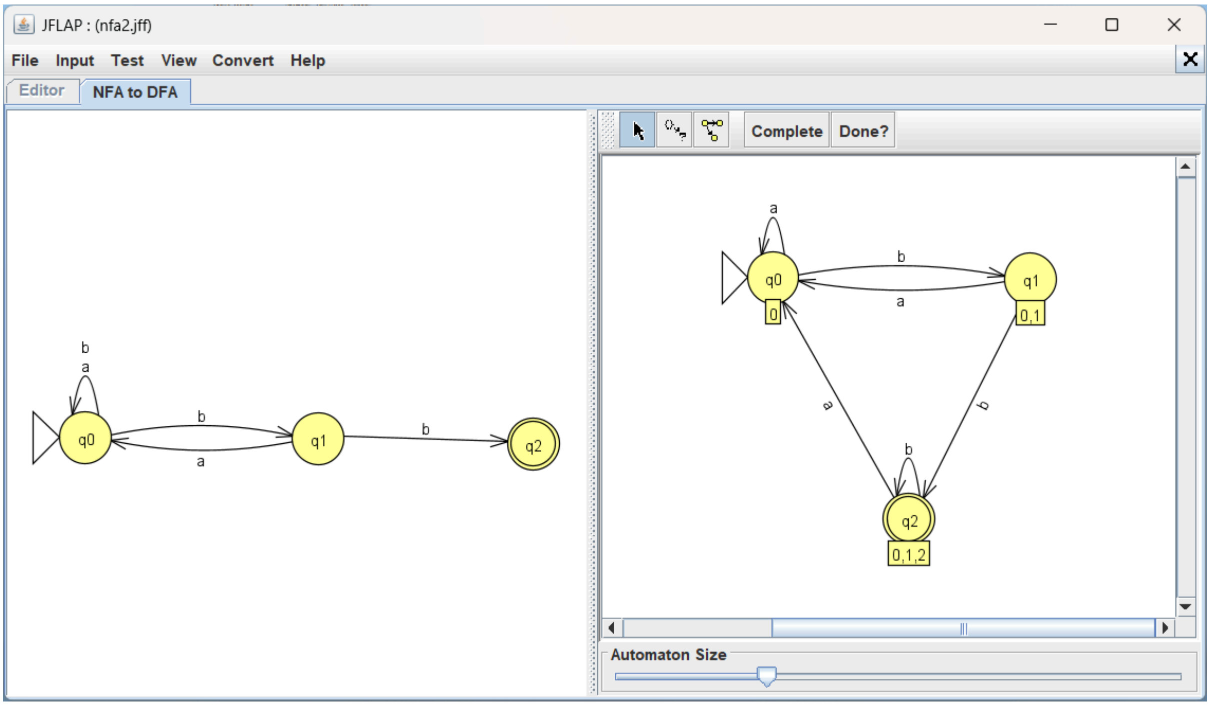

Figura 27: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{C}, {A, B}}
π' = {{C}, {A}{B}}

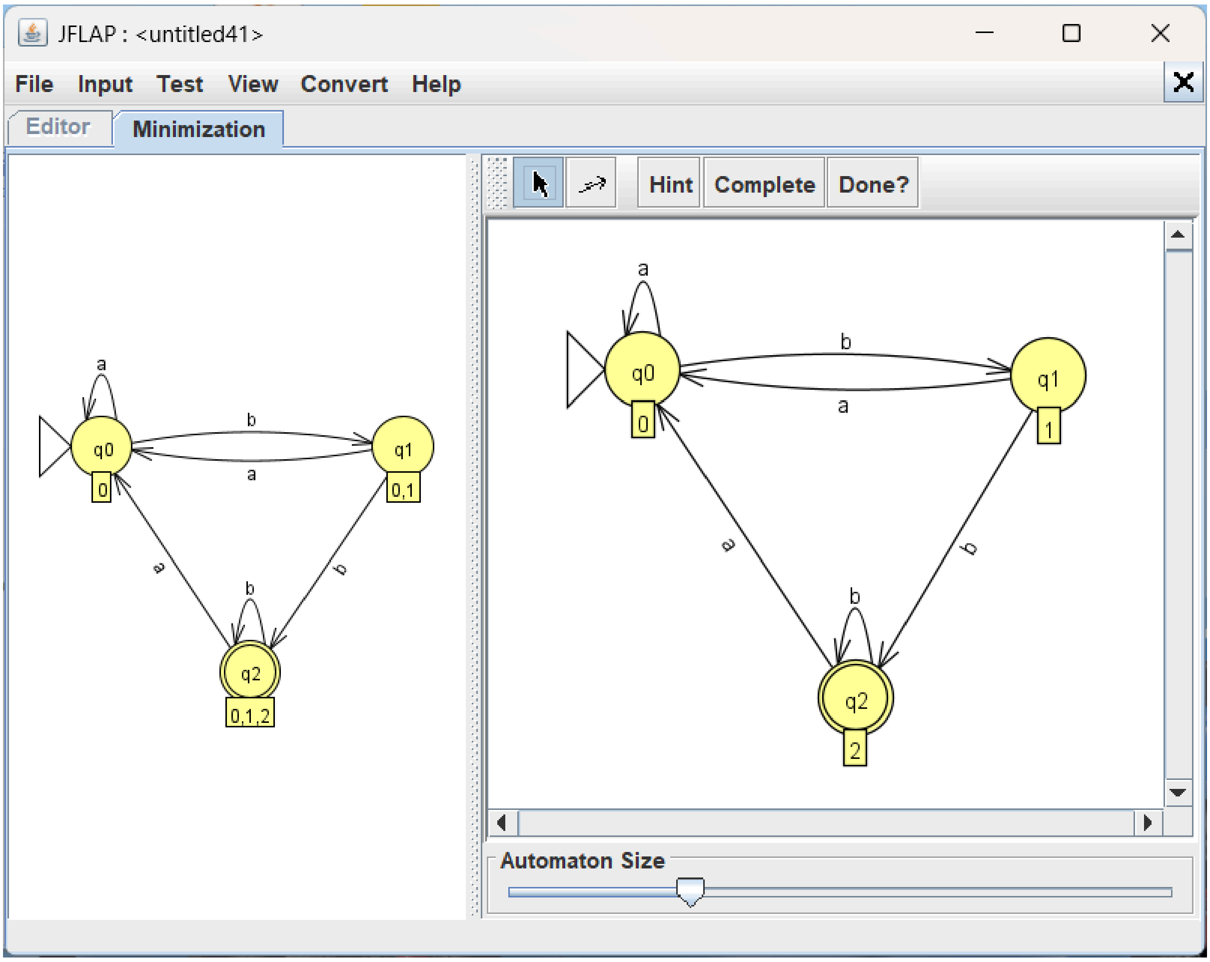

Figura 28: DFA mínimo equivalente.

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

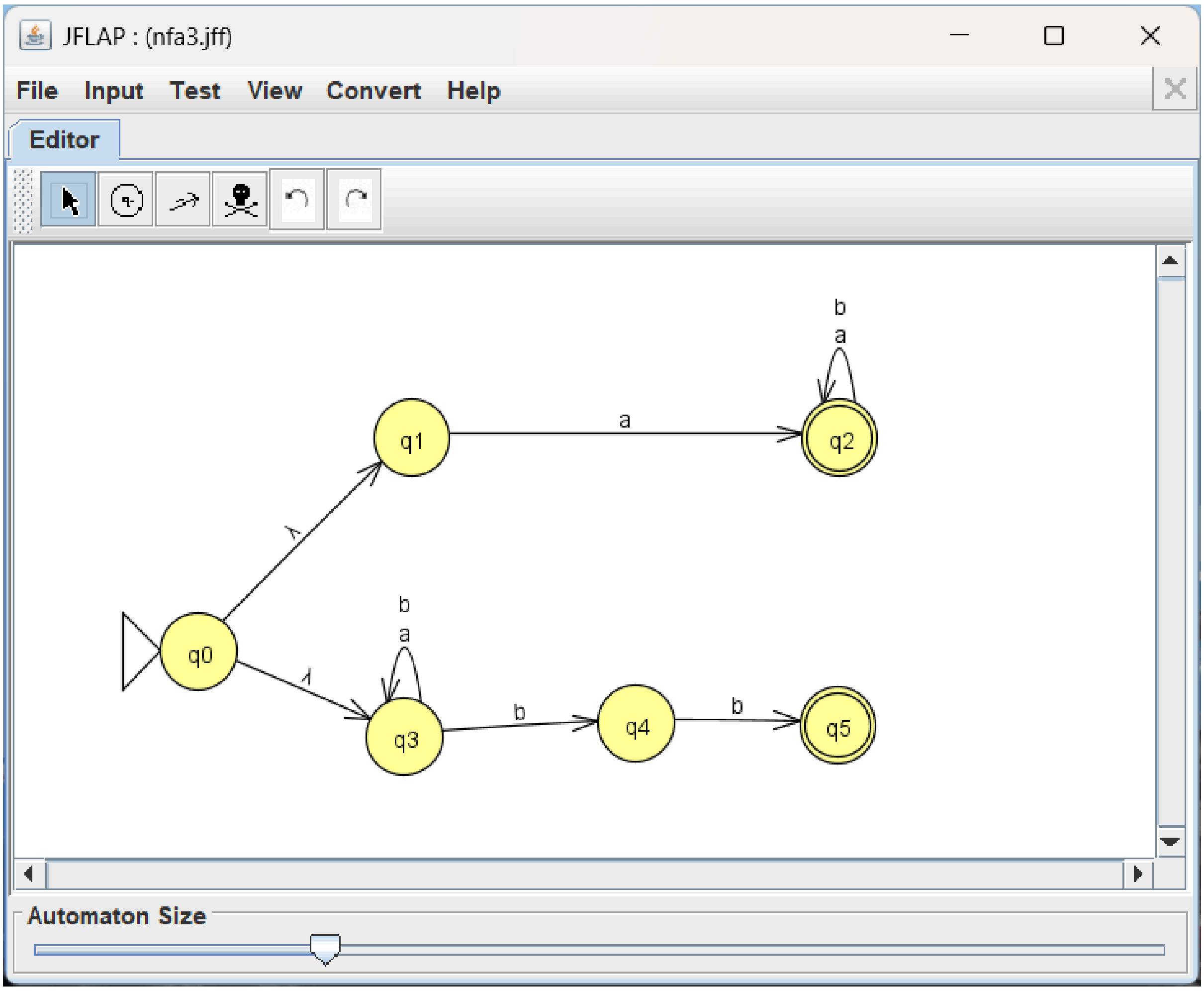

Figura 29: NFA que reconoce cadenas que que empiecen por “a” o terminen en “bb” sobre el alfabeto Σ = {a, b}

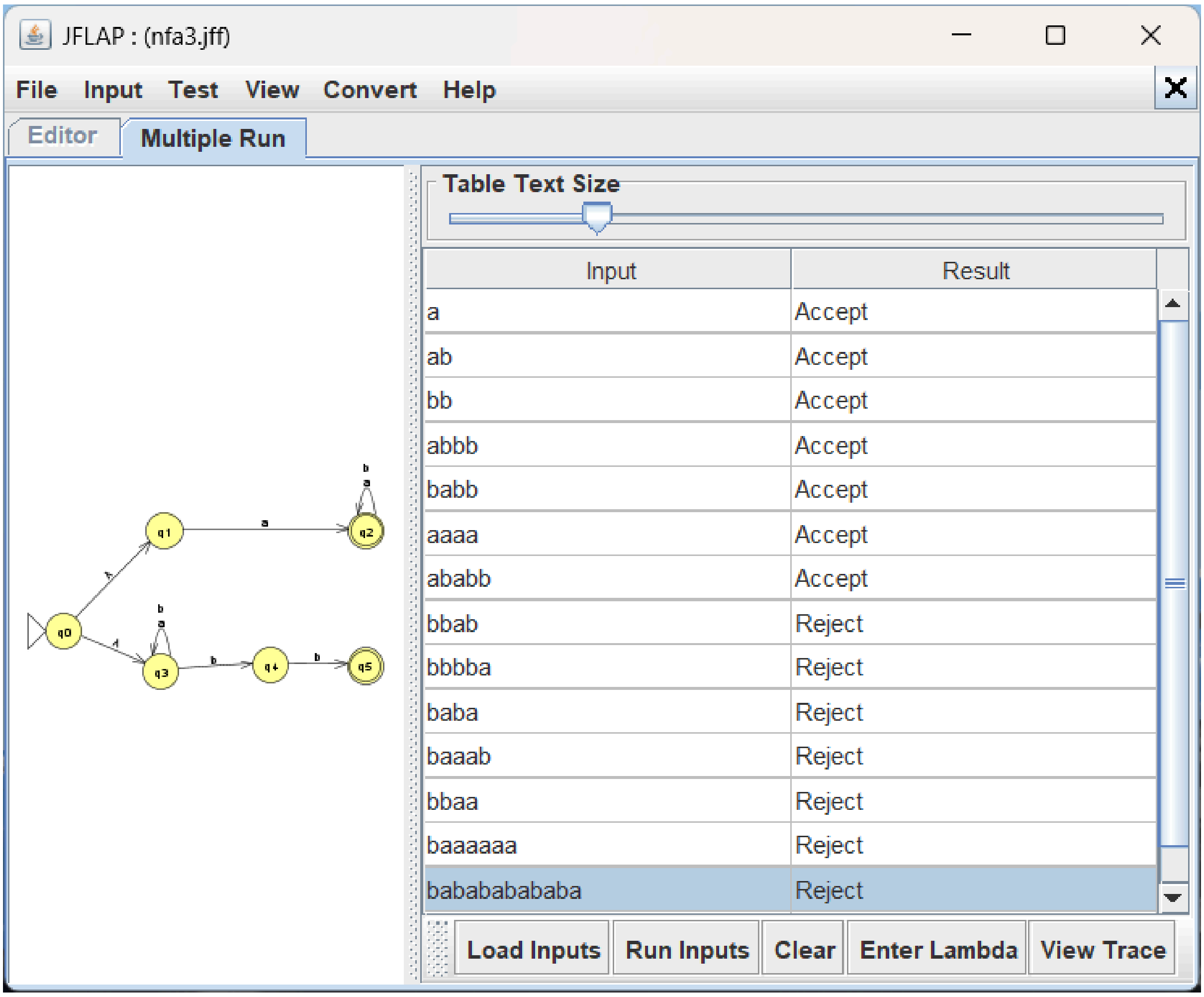

Figura 30: Cadenas de prueba para el NFA anterior.

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

ϵ-clausura({q0}) = {q0, q1, q3} = A
δ(A, a) = {q2, q3} = B
δ(A, b) = {q3, q4} = C
ϵ-clausura(B) = {q2, q3} = B
δ(B, a) = {q2, q3} = B
δ(B, b) = {q2, q3, q4} = D
ϵ-clausura(C) = {q3, q4} = C
δ(C, a) = { q3} = E
δ(C, b) = { q3, q4, q5} = F
ϵ-clausura(D) = {q2, q3, q4} = D
δ(D, a) = { q2, q3} = B
δ(D, b) = { q2, q3, q4, q5} = G
ϵ-clausura(E) = {q3} = E
δ(E, a) = {q3} = E

δ(E, b) = {q3, q4} = C
ϵ-clausura(F) = {q3, q4, q5} = F
δ(F, a) = {q3} = E
δ(F, b) = {q3, q4, q5} = F
ϵ-clausura(G) = {q2, q3, q4, q5} = G
δ(G, a) = {q2, q3} = E
δ(G, b) = {q2, q3, q4, q5} = G

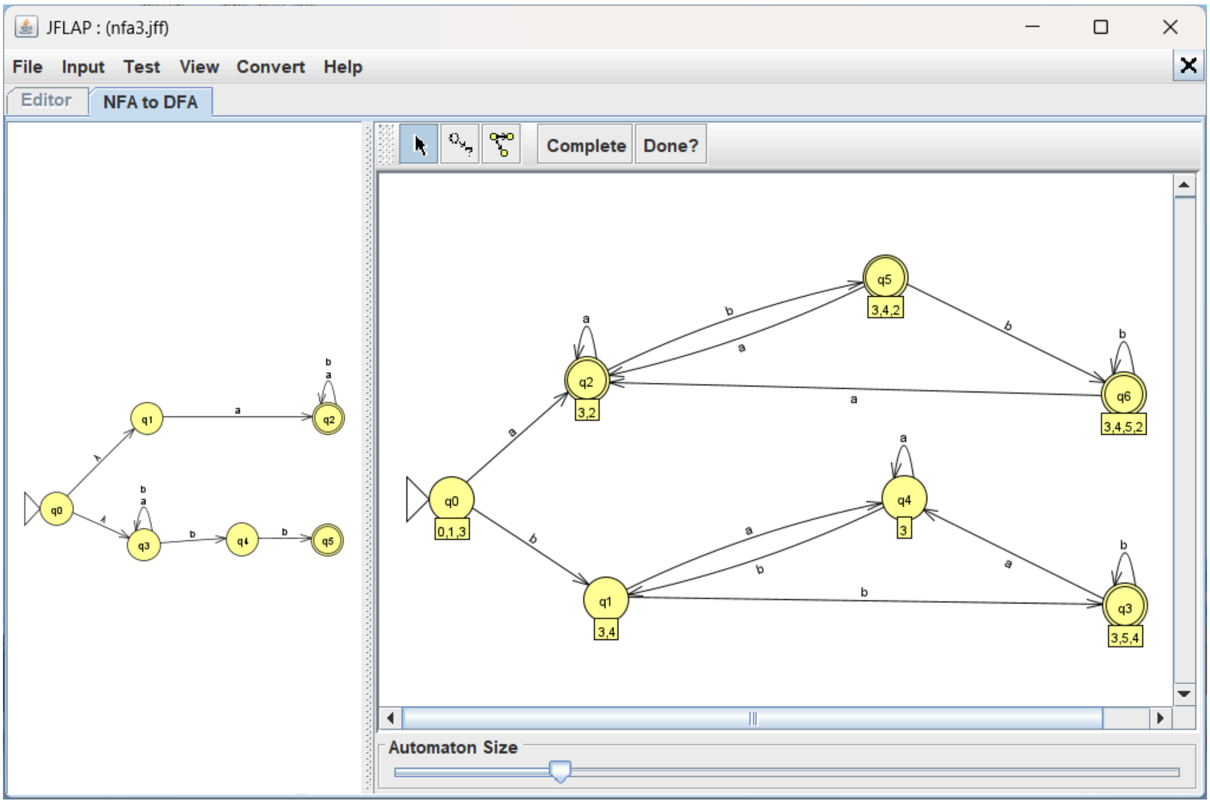

Figura 31: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{B, D, G, F}, {A, C, E}}
π' = {{B, D, G, F}, {A}, {C, E}}
π'' = {{B, D, G}, {F}, {A}, {C}, {E}}

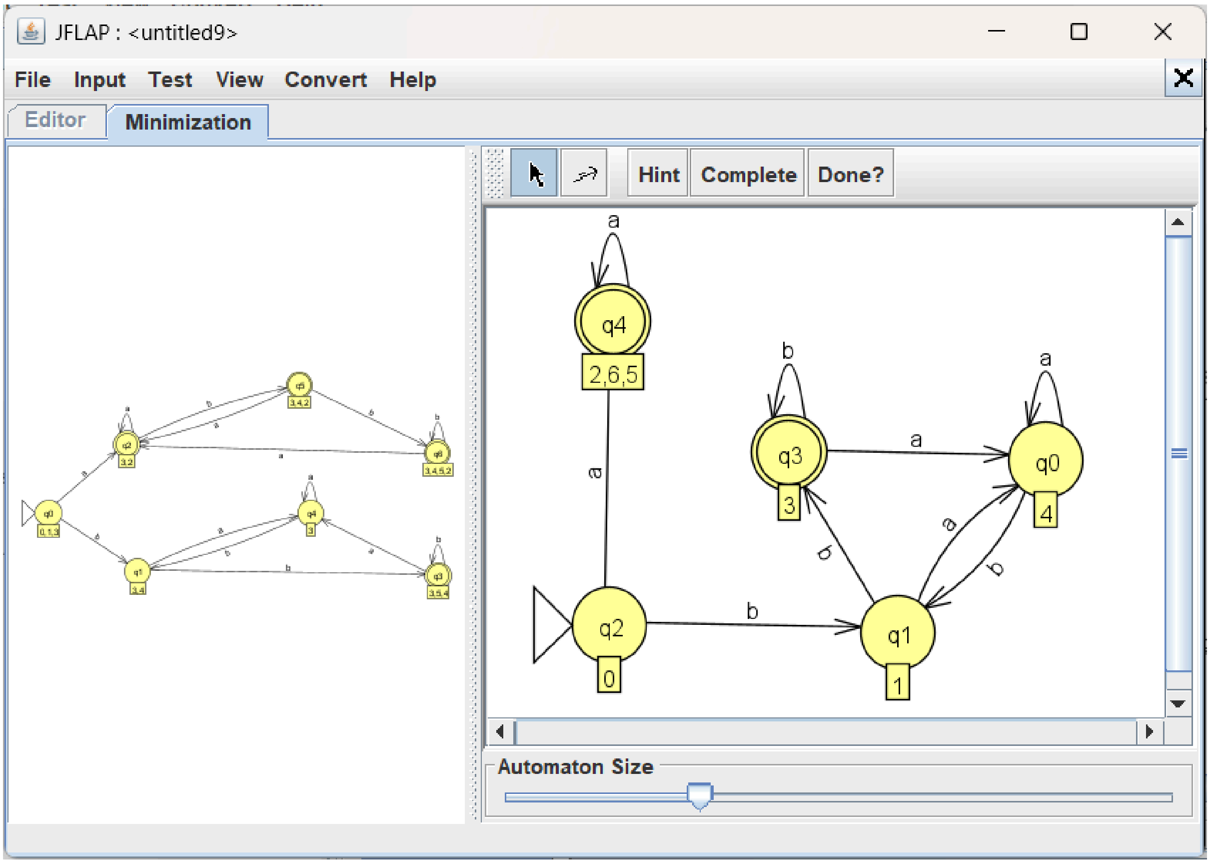

Figura 32: DFA mínimo equivalente.

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

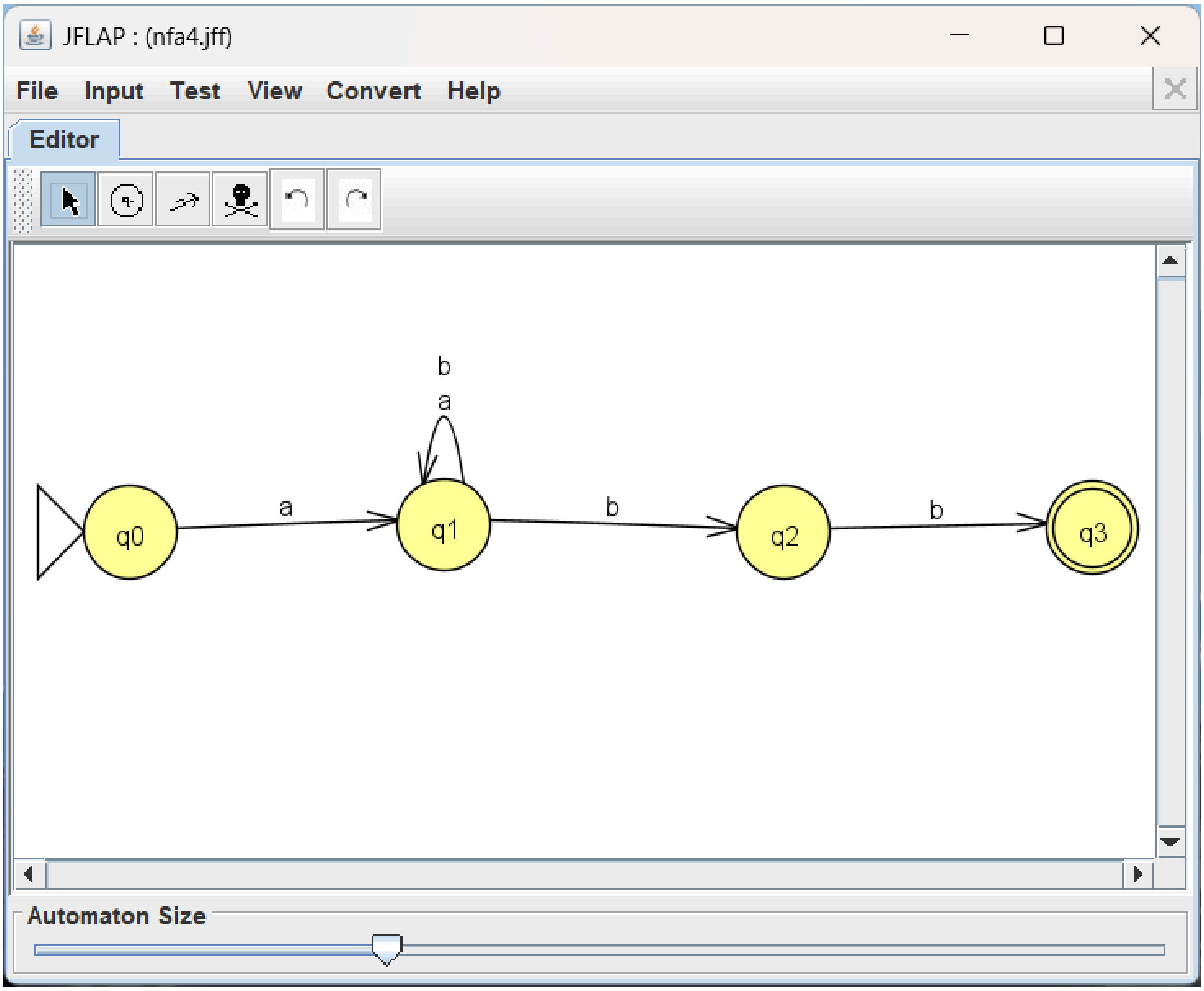

Figura 33: NFA que reconoce cadenas que empiecen por "a" y terminen en "bb" sobre el alfabeto Σ = {a, b}

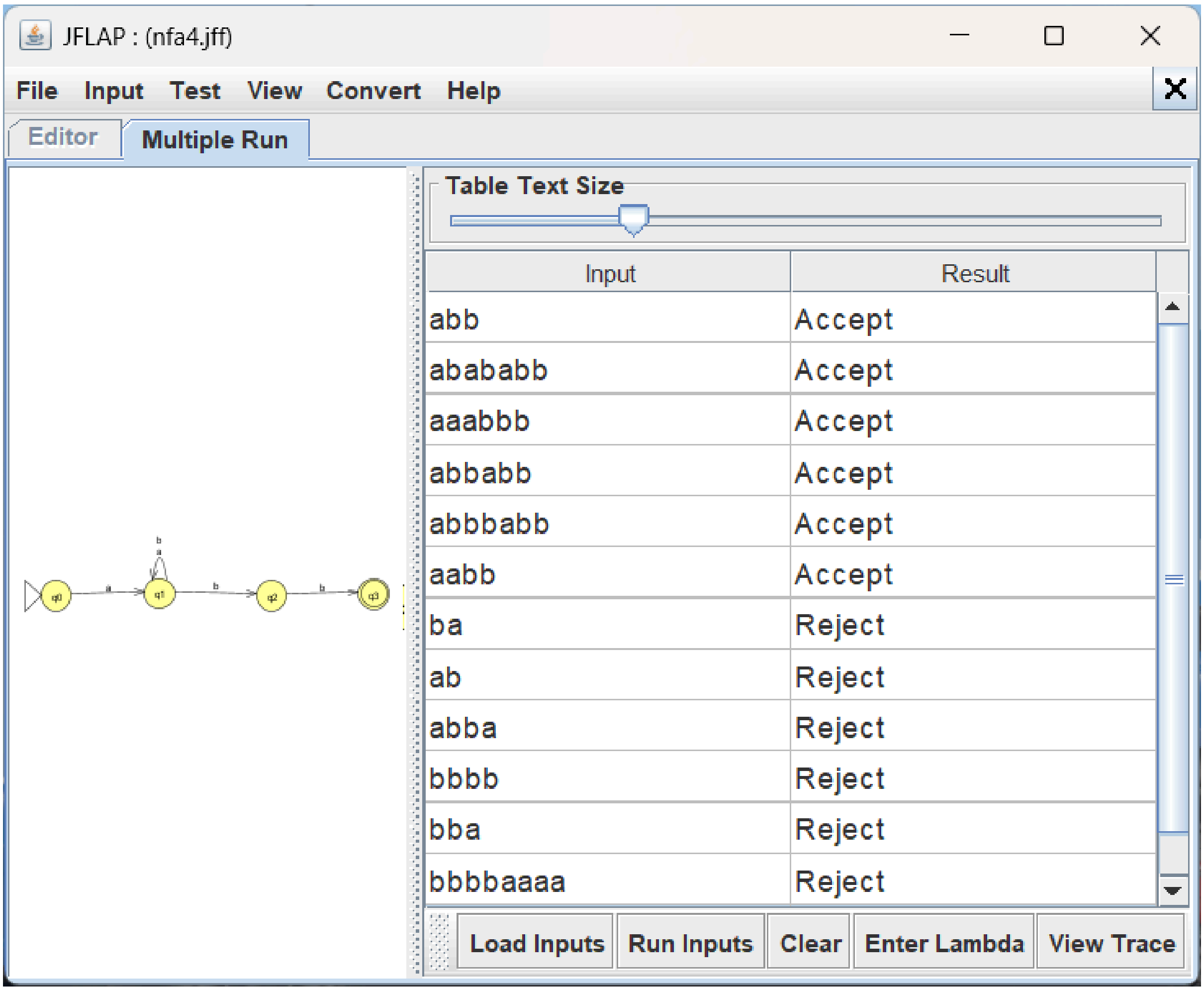

Figura 34: Cadenas de prueba para el NFA anterior.

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

$S = \{q0\} = A$
$\delta(A, a) = \{q1\} = B$
$\delta(A, b) = \varnothing$
$\delta(B, a) = \{q1\} = B$
$\delta(B, b) = \{q1, q2\} = C$
$\delta(C, a) = \{q1\} = B$
$\delta(C, b) = \{q1, q2, q3\} = D$
$\delta(D, a) = \{q1\} = B$
$\delta(D, b) = \{q1, q2, q3\} = D$

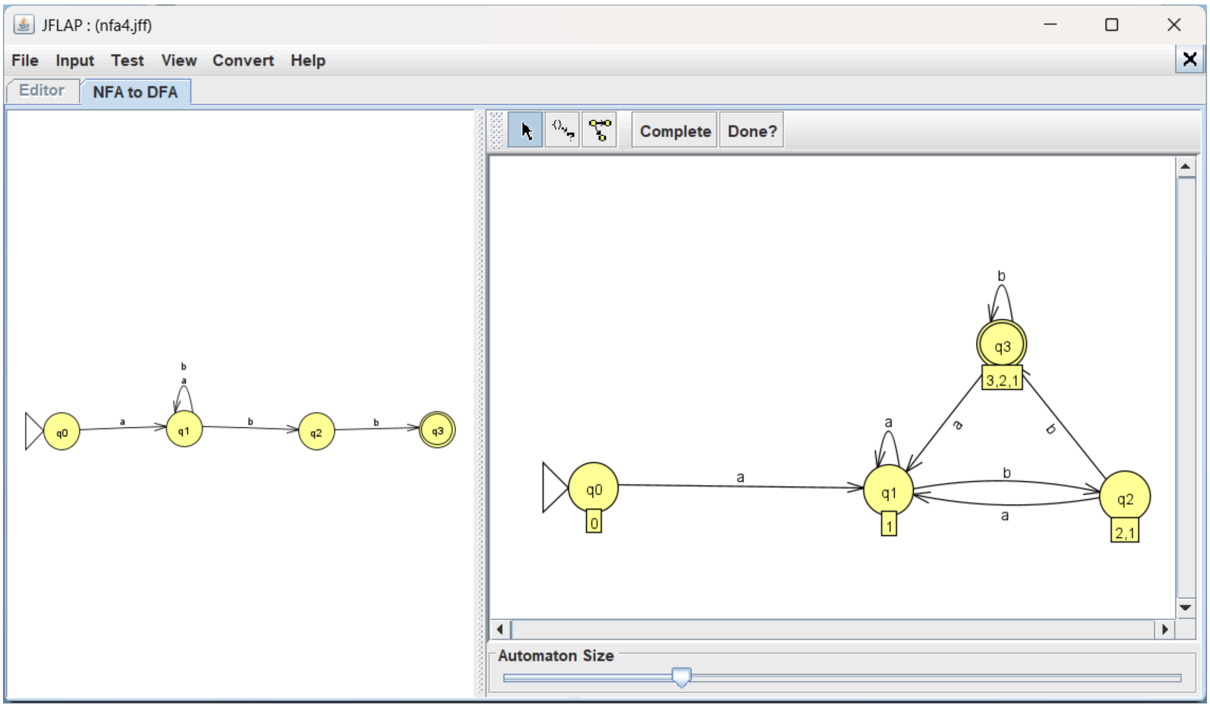

Figura 35: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{D}, {A, B, C}}
π' = {{D}, {A}, {B, C}}
π'' = {{D}, {A}, {B}, {C}}

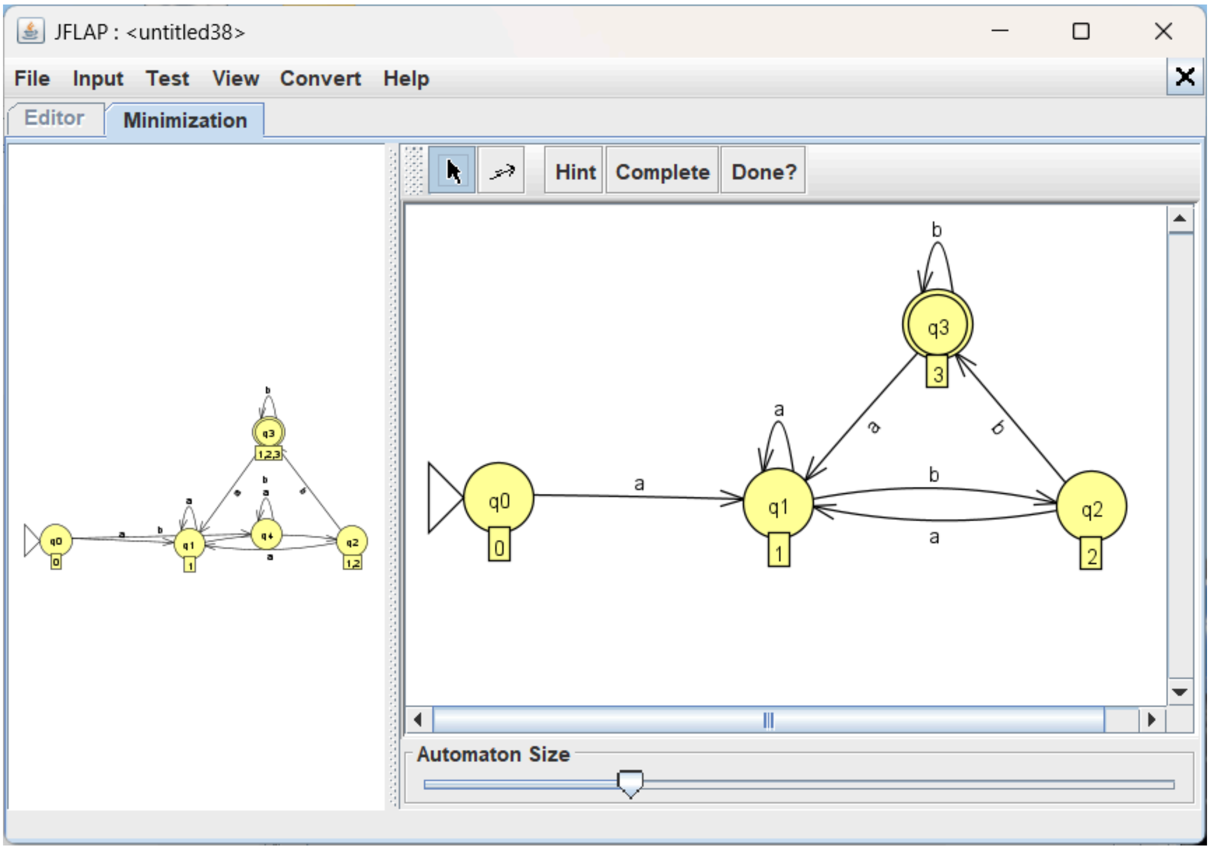

Figura 36: DFA mínimo equivalente.

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

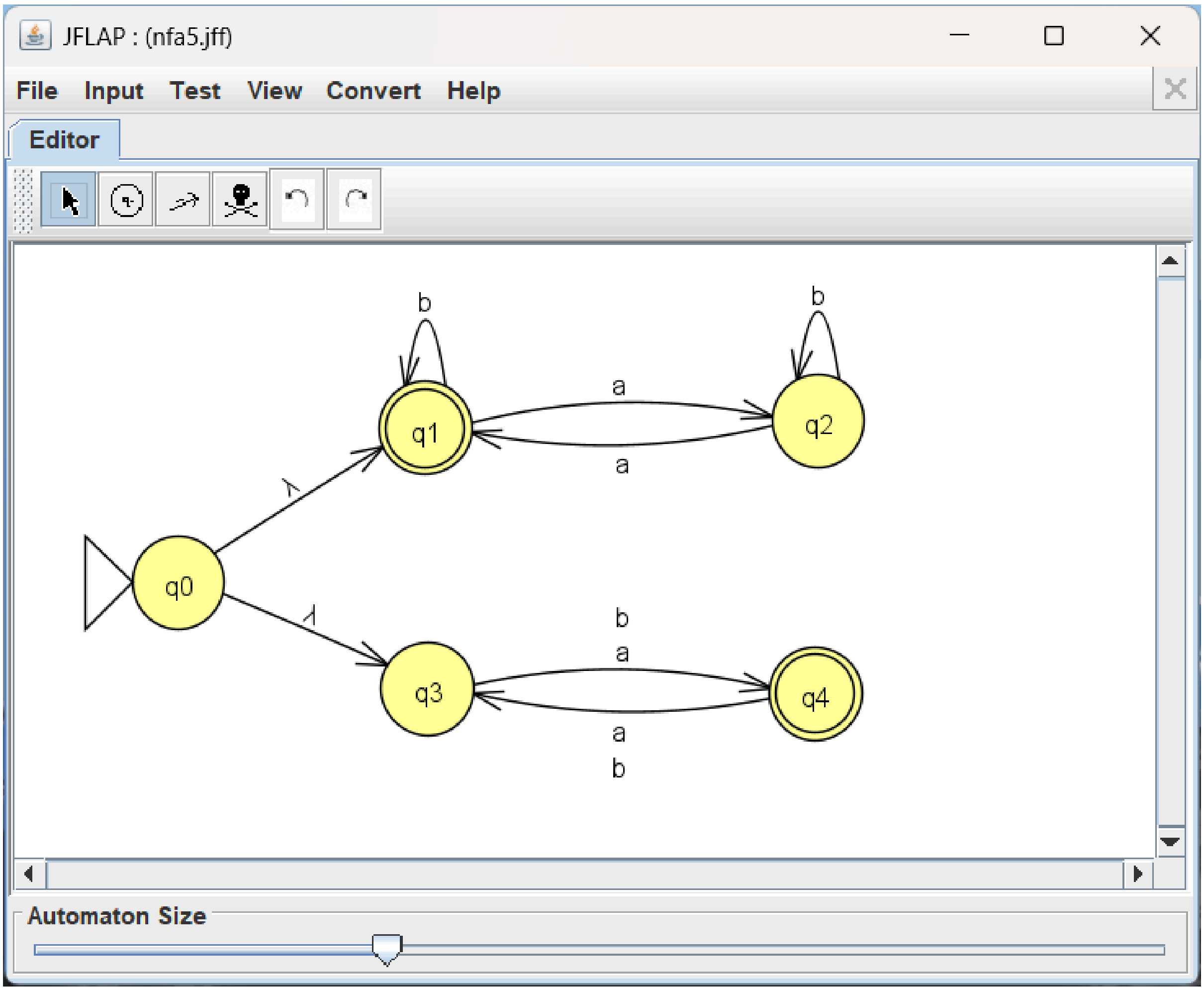

Figura 37: NFA que reconoce cadenas con número de “a’s” par o longitud impar sobre el alfabeto Σ = {a, b}

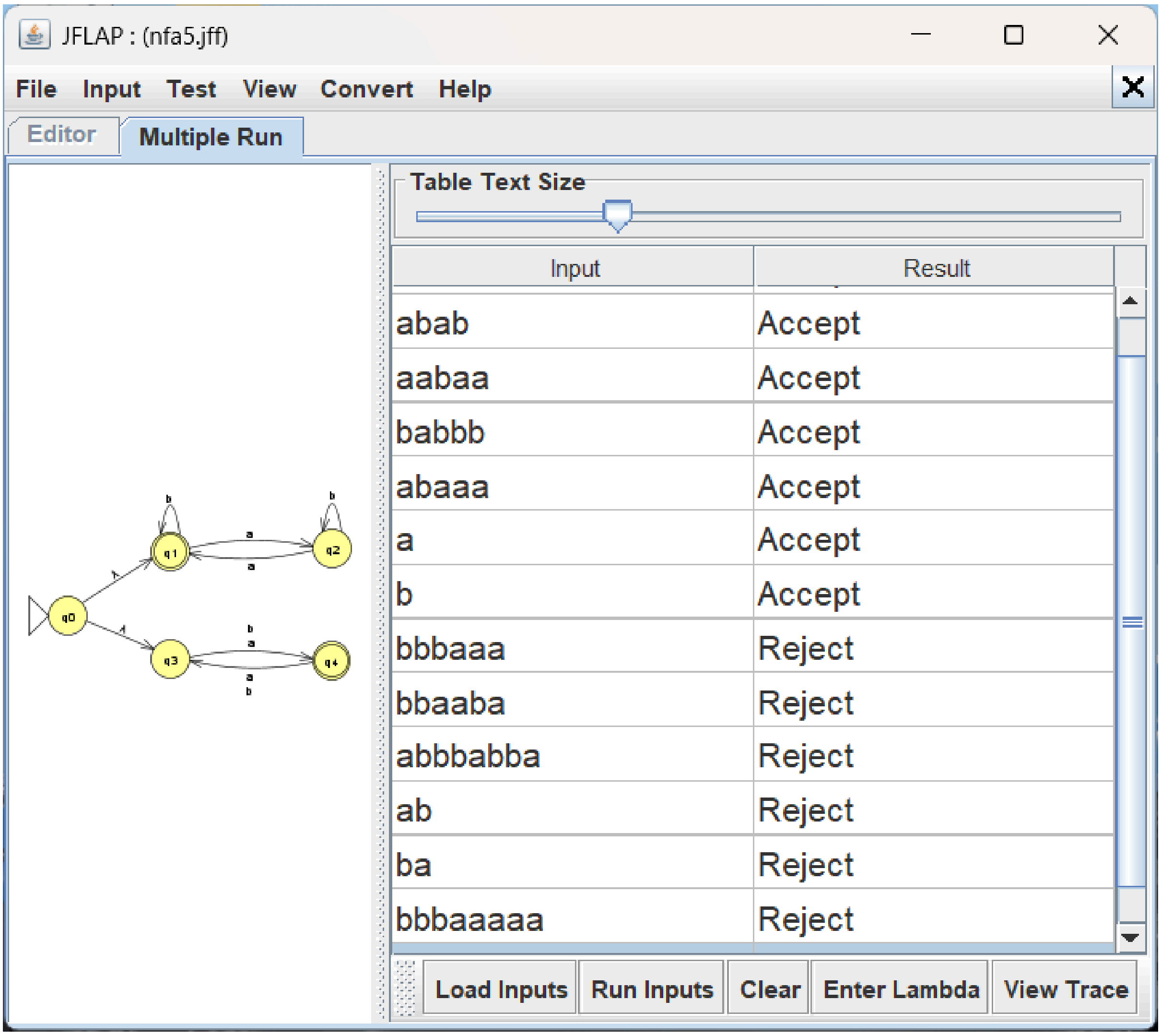

Figura 38: Cadenas de prueba para el NFA anterior.

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

ϵ-clausura({q0}) = {q0, q1, q3} = A
δ(A, a) = {q2, q4} = B
δ(A, b) = {q1, q4} = C
ϵ-clausura(B) = {q2, q4} = B
δ(B, a) = {q1, q3} = D
δ(B, b) = {q2, q3} = E
ϵ-clausura(C) = {q1, q4} = C
δ(C, a) = {q2, q3} = E
δ(C, b) = {q1, q3} = D
ϵ-clausura(D) = {q1, q3} = D
δ(D, a) = {q2, q4} = B
δ(D, b) = {q1, q4} = C

ϵ-clausura(E) = {q2, q3} = E
δ(E, a) = {q1, q4} = C
δ(E, b) = {q2, q4} = B

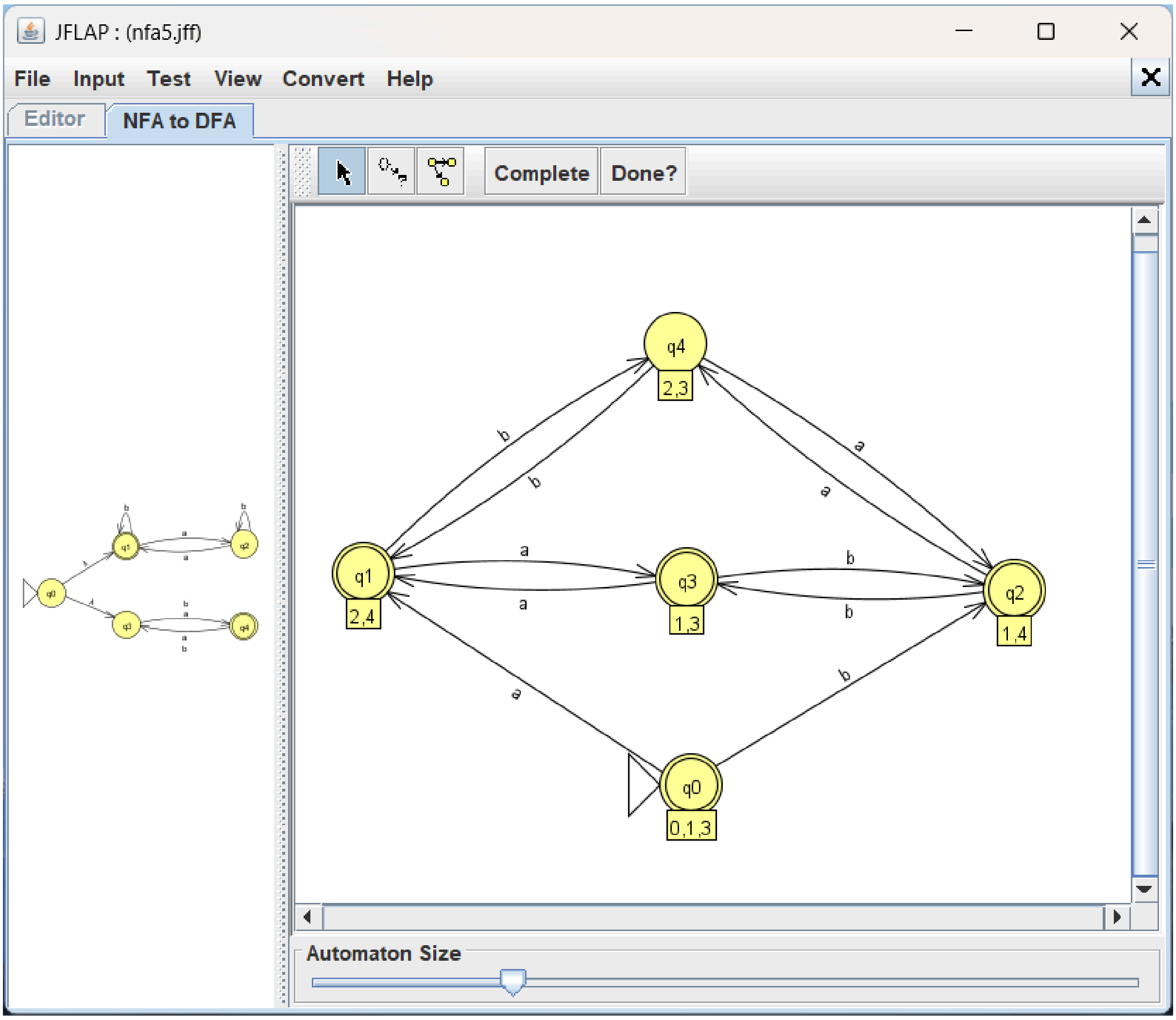

Figura 39: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{A, B, C, D}, {E}}
π' = {{A, B, D}, {C}, {E}}
π'' = {{A}, {B, D}, {C}, {E}}

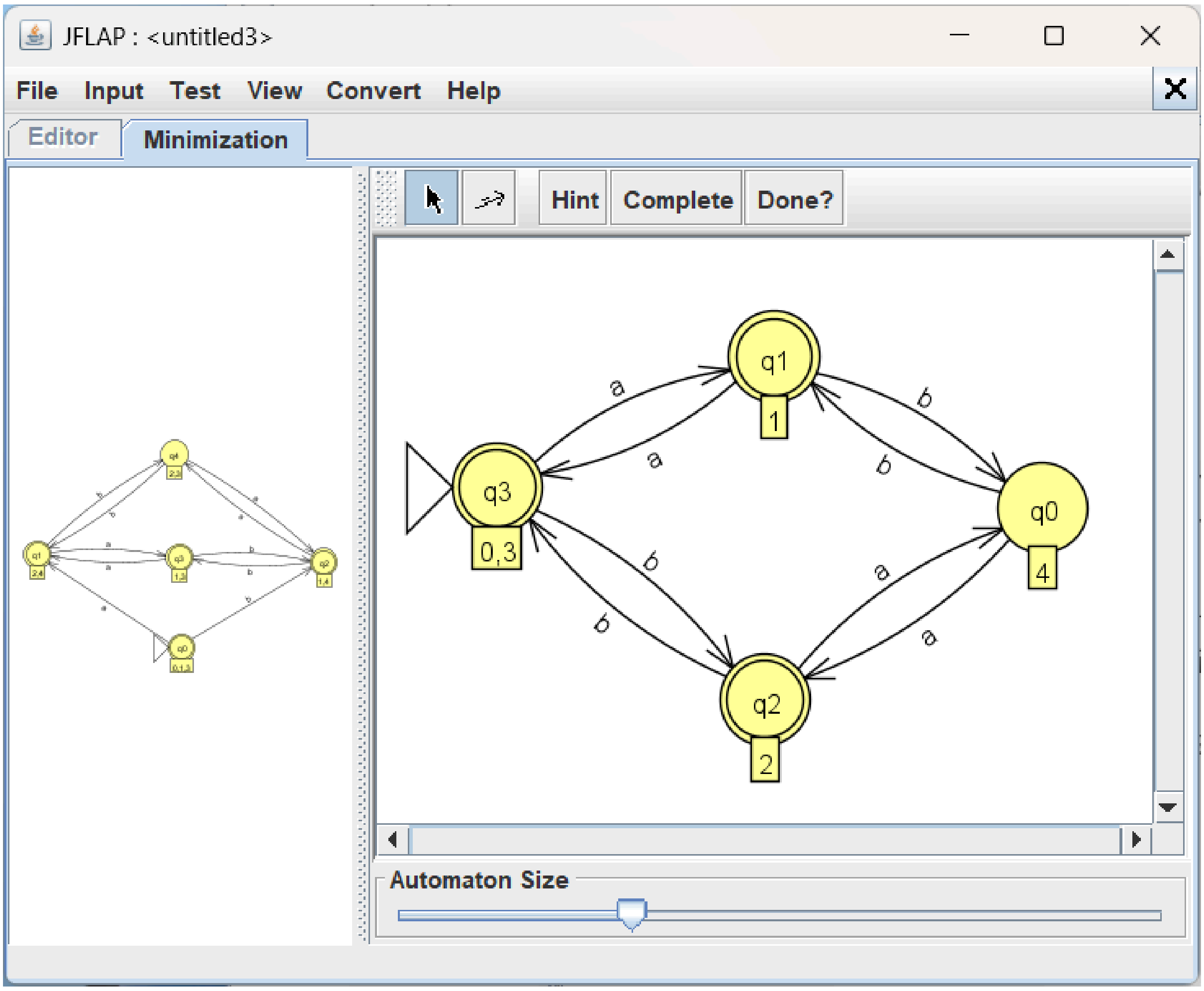

Figura 40: DFA mínimo equivalente.

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

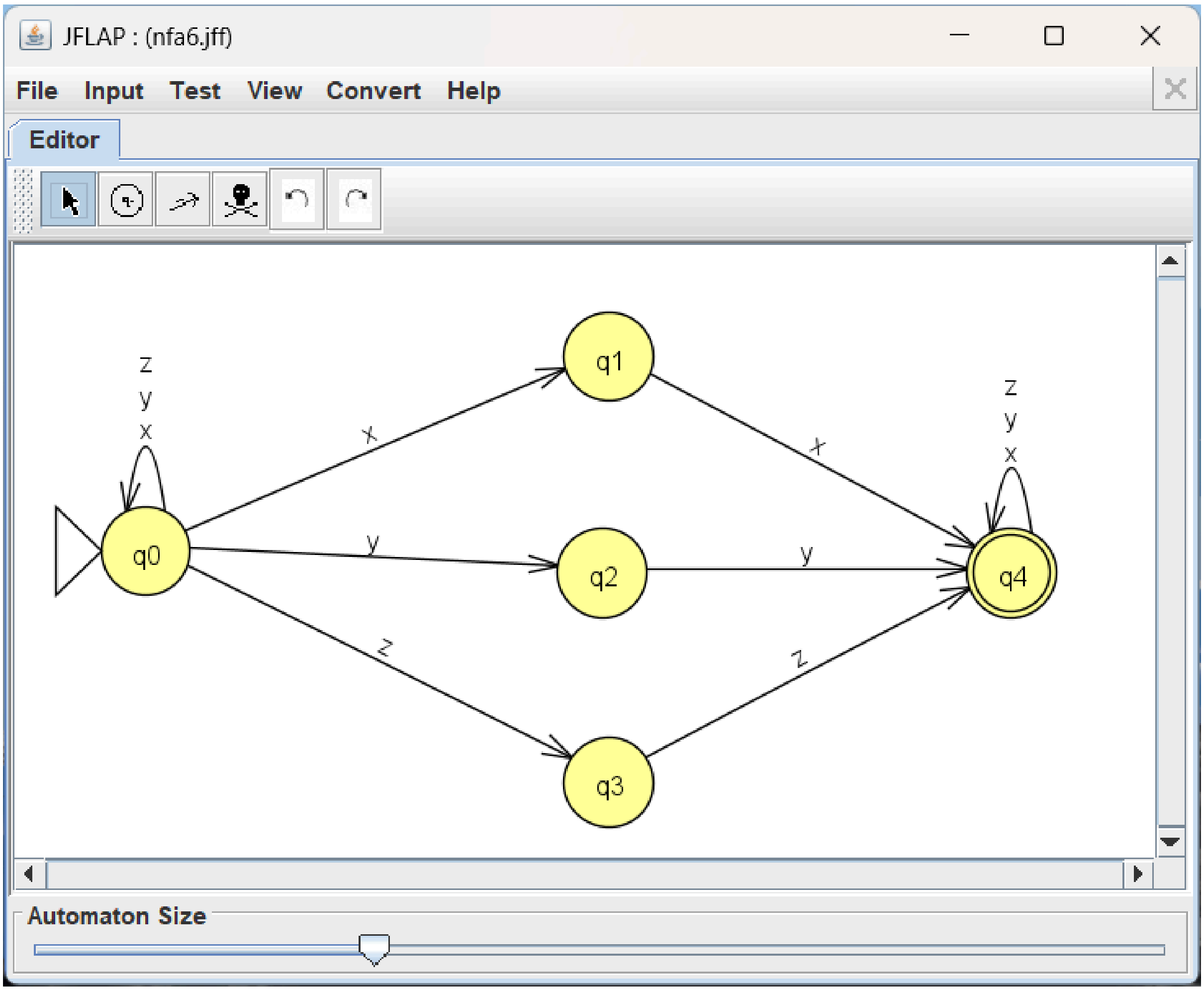

Figura 41: NFA que reconoce cadenas que contengan al menos dos símbolos iguales consecutivos sobre el alfabeto Σ = {x, y, z}

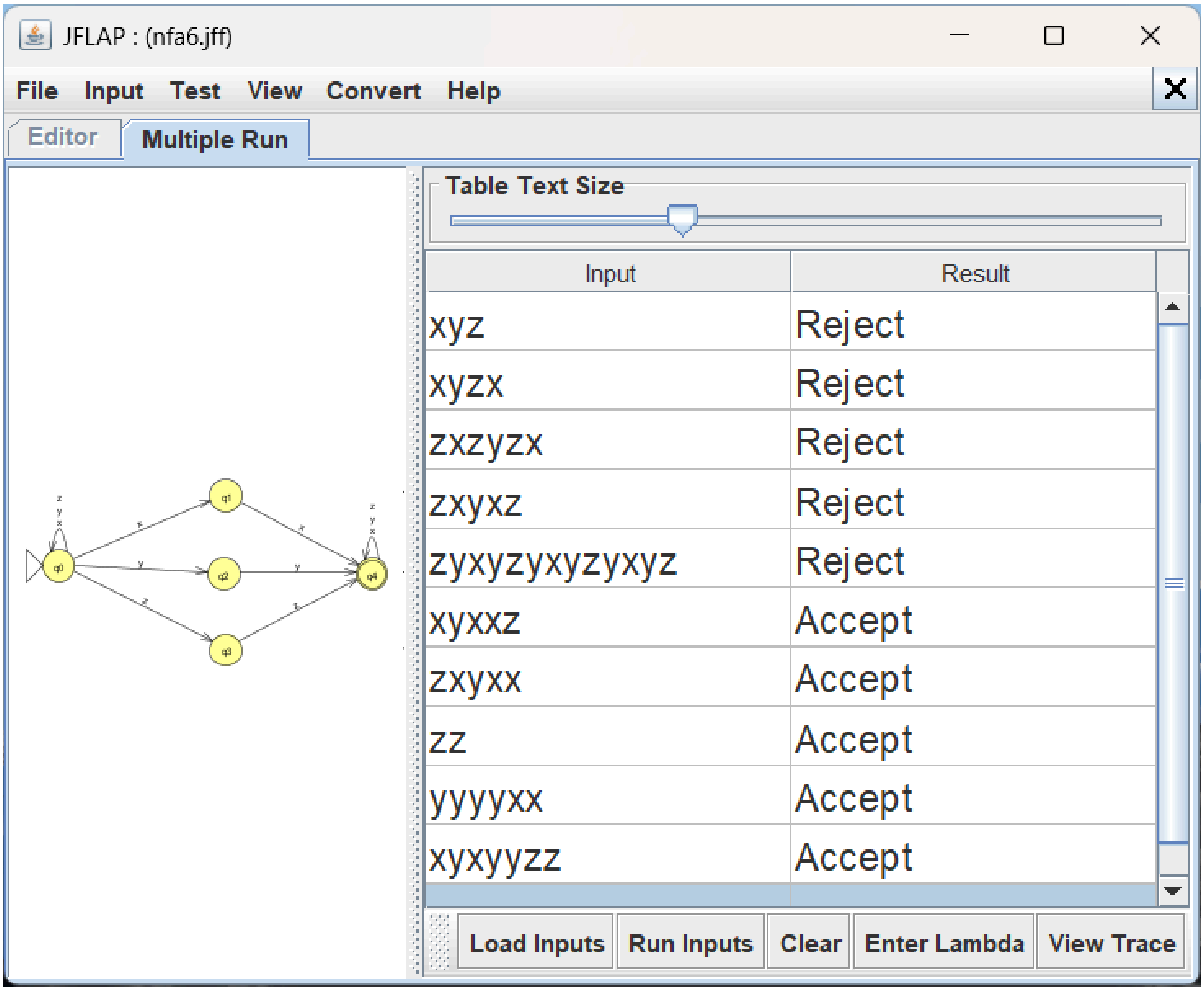

Figura 42: Cadenas de prueba para el NFA anterior.

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

S = {q0} = A
δ(A, x) = {q0, q1} = B
δ(A, y) = {q0, q2} = C
δ(A, z) = {q0, q3} = D
δ(B, x) = {q0, q1, q4} = E
δ(B, y) = {q0, q2} = C
δ(B, z) = {q0, q3} = D
δ(C, x) = {q0, q1} = B
δ(C, y) = {q0, q2, q4} = F
δ(C, z) = {q0, q3} = D
δ(D, x) = {q0, q1} = B
δ(D, y) = {q0, q2} = C
δ(D, z) = {q0, q3, q4} = G
δ(E, x) = {q0, q1, q4} = E
δ(E, y) = {q0, q2, q4} = F

δ(E, z) = {q0, q3, q4} = G
δ(F, x) = {q0, q1, q4} = E
δ(F, y) = {q0, q2, q4} = F
δ(F, z) = {q0, q3, q4} = G
δ(G, x) = {q0, q1, q4} = E
δ(G, y) = {q0, q2, q4} = F
δ(G, z) = {q0, q3, q4} = G

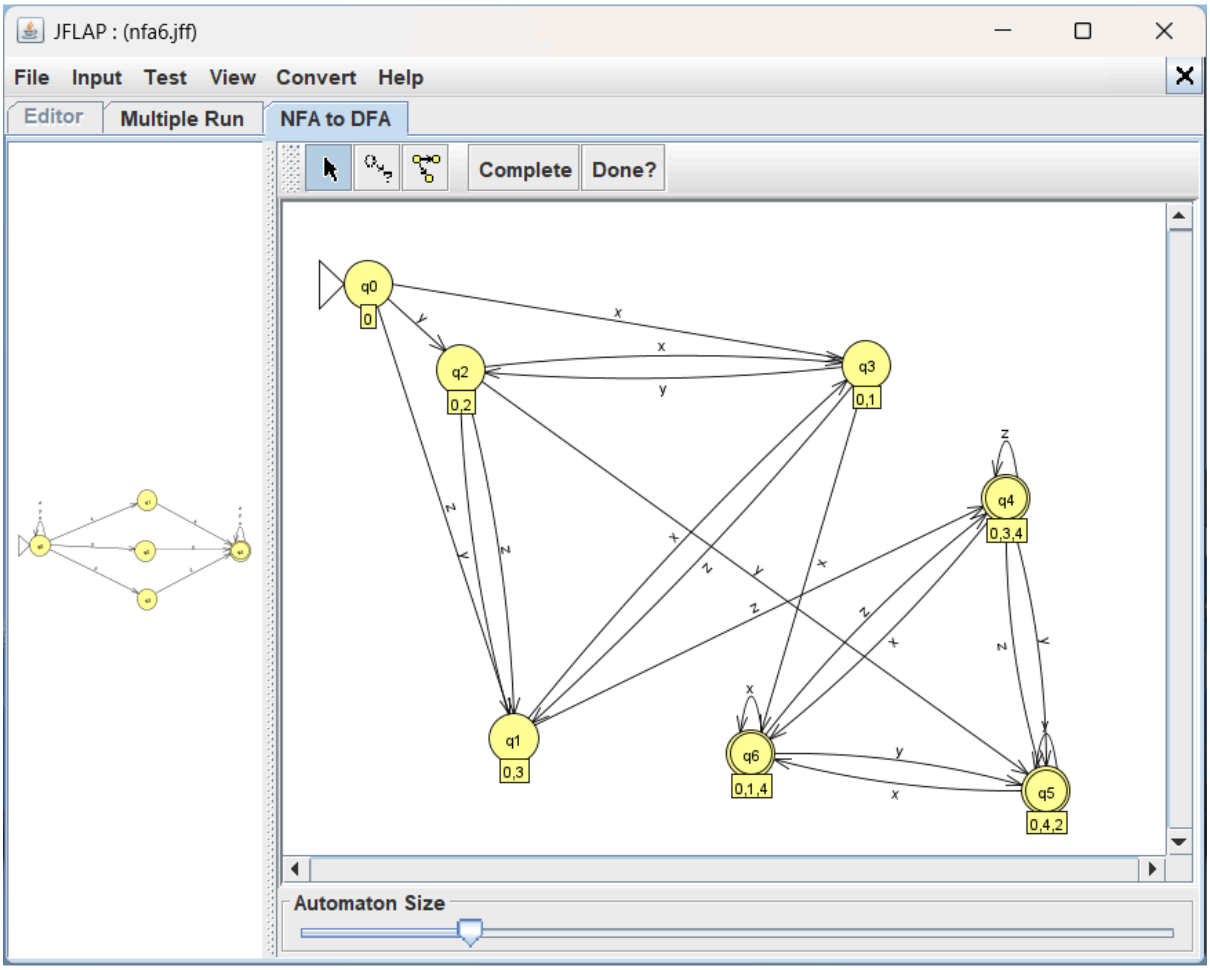

Figura 43: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{E, F, G}, {A, B, C, D}}
π' = {{E, F, G}, {B}, {A, C, D}}
π'' = {{E, F, G}, {B}, {C}, {A, D}}
π''' = {{E, F, G}, {B}, {C}, {D}, {A}}

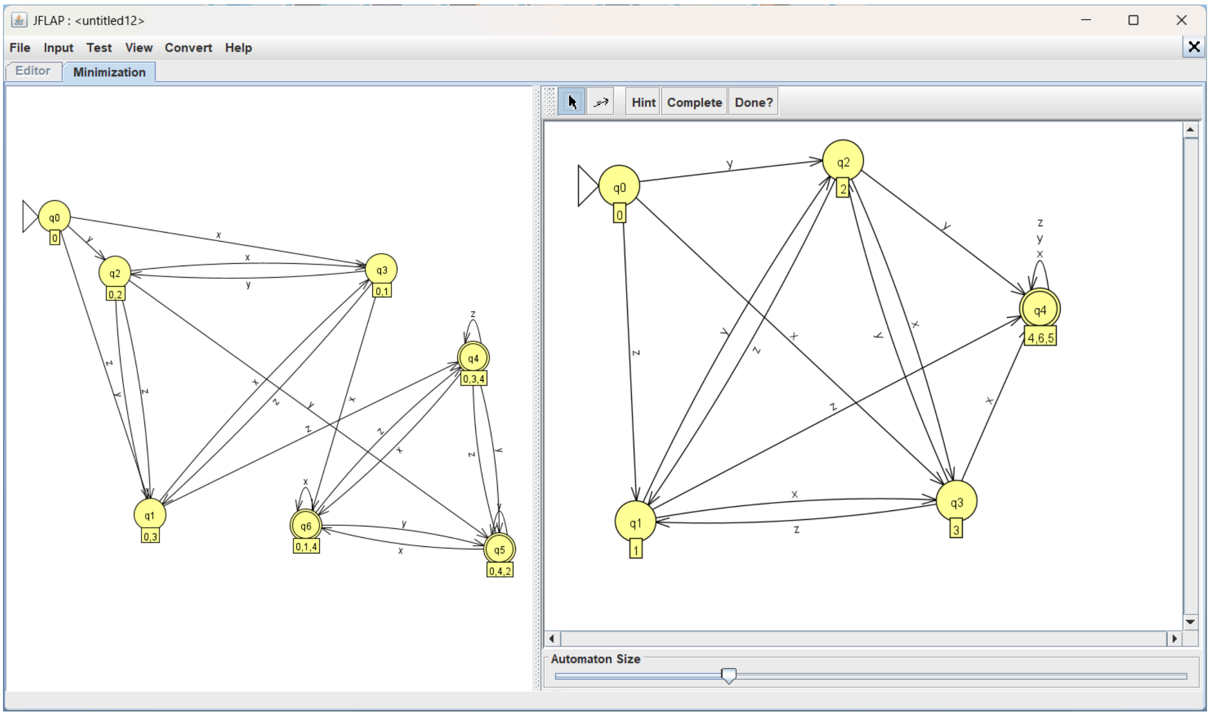

Figura 44: DFA mínimo equivalente.

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

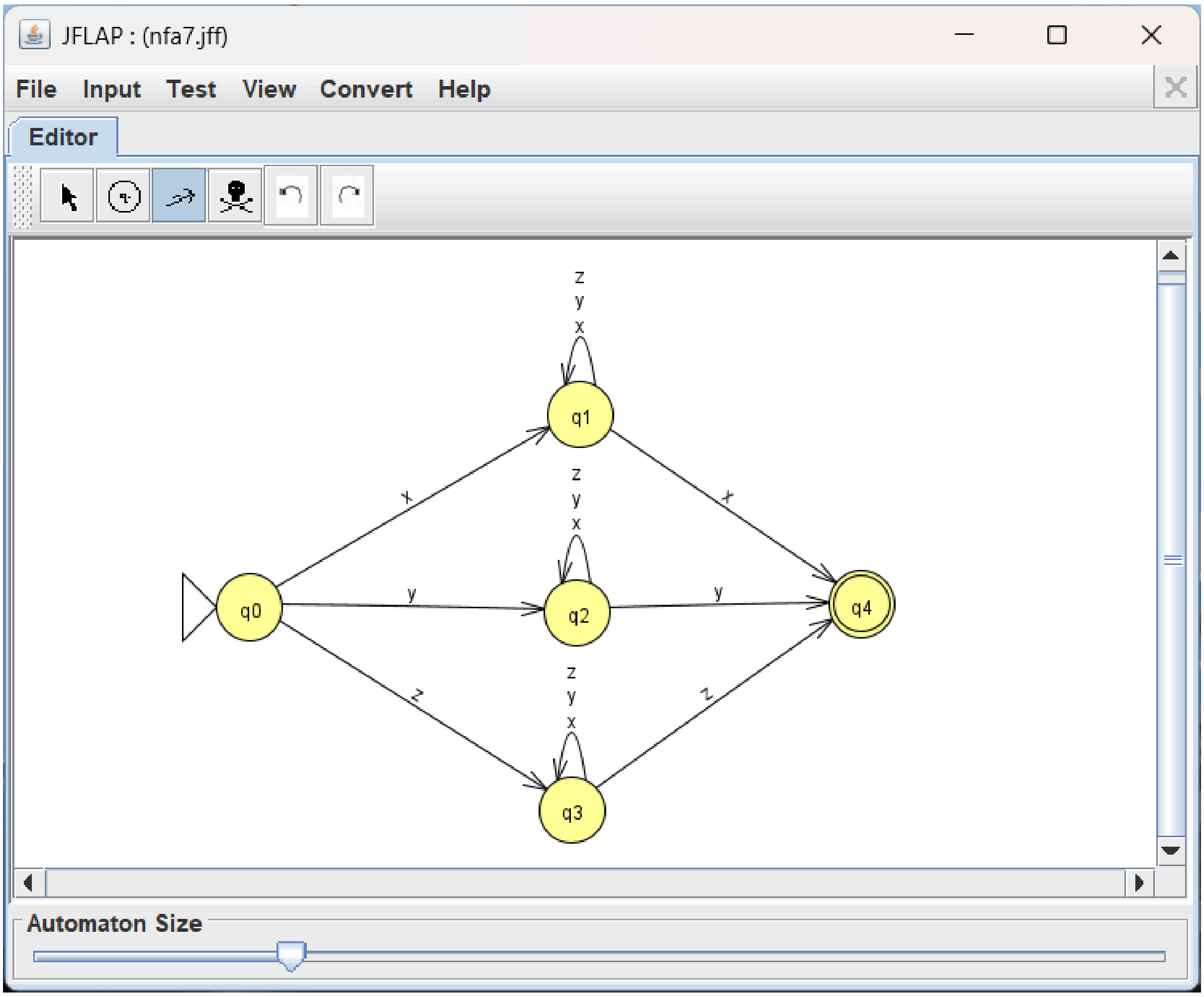

Figura 45: NFA que reconoce cadenas w sobre el alfabeto {x, y, z}, con |w| ≥ 2, tales que w empieza y termina por el mismo símbolo.

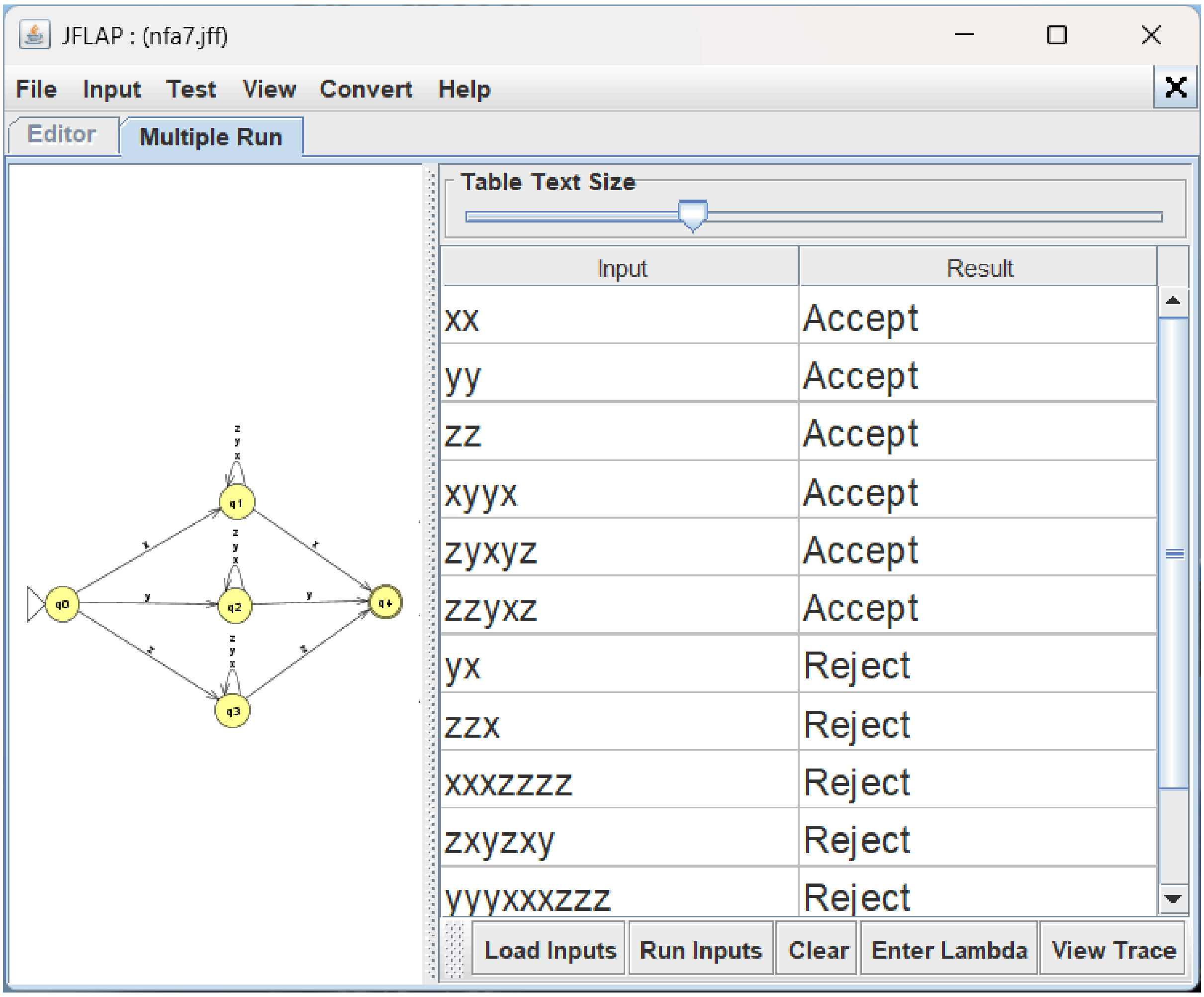

Figura 46: Cadenas de prueba para el NFA anterior.

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

S = {q0} = A
δ(A, x) = {q1} = B
δ(A, y) = {q2} = C
δ(A, z) = {q3} = D
δ(B, x) = {q1, q4} = E
δ(B, y) = {q1} = B
δ(B, z) = {q1} = B
δ(C, x) = {q2} = C
δ(C, y) = {q2, q4} = F
δ(C, z) = {q2} = C
δ(D, x) = {q3} = D
δ(D, y) = {q3} = D
δ(D, z) = {q3, q4} = G
δ(E, x) = {q1, q4} = E

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

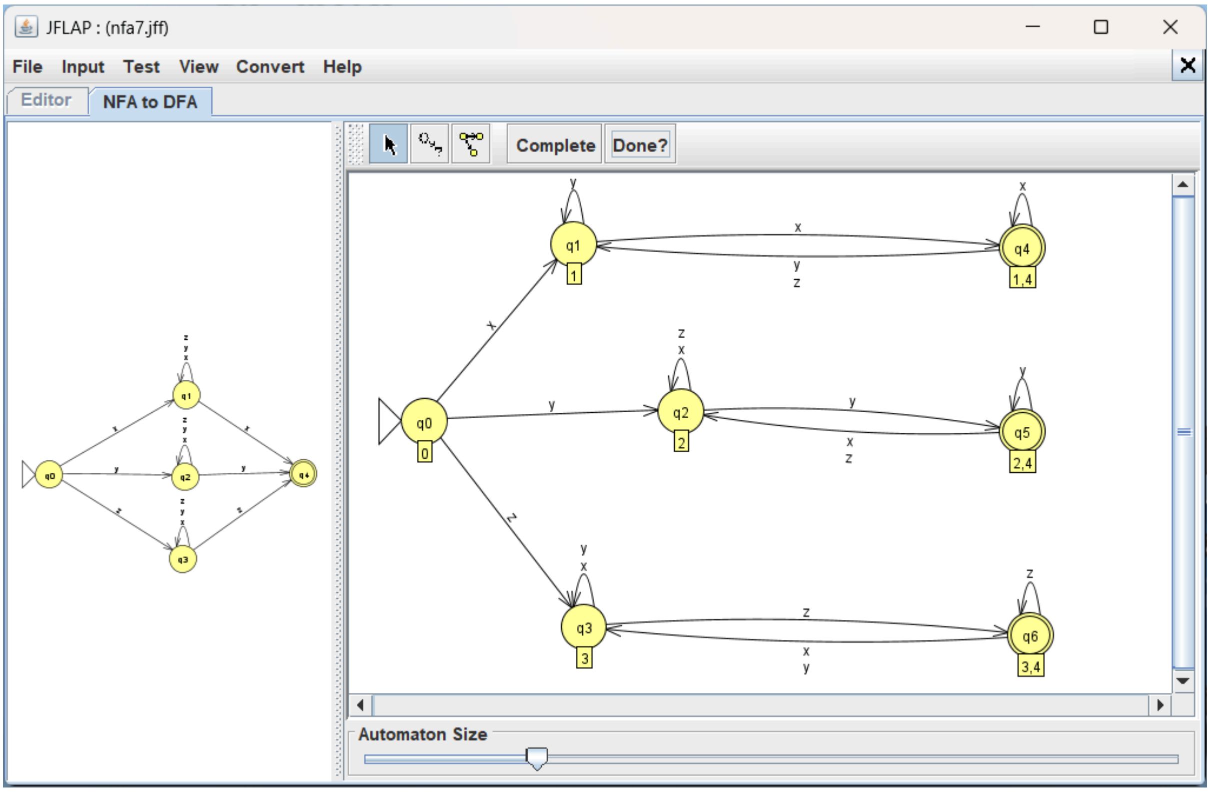

Figura 47: DFA equivalente al NFA anterior.

Algoritmo de minimización de estados para obtener el DFA mínimo:

π = {{E, F, G}, {A, B, C, D}}
π' = {{E, F, G}, {B}, {A, C, D}}
π'' = {{E, F, G}, {B}, {C}, {A, D}}
π''' = {{E, F, G}, {B}, {C}, {D}, {A}}
π'''' = {{E, G}, {F}, {B}, {C}, {D}, {A}}
π''''' = {{E}, {G}, {F}, {B}, {C}, {D}, {A}}

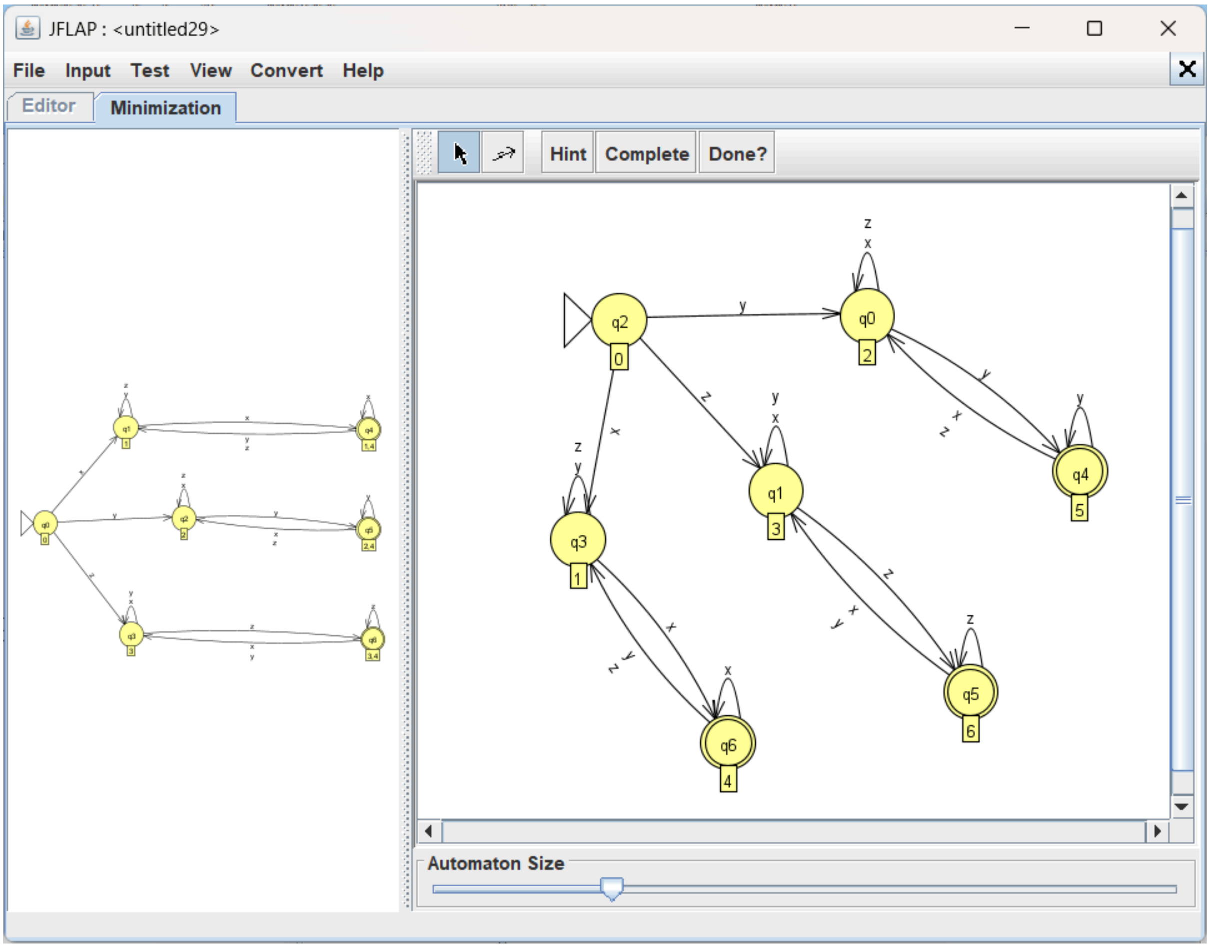

Figura 48: DFA mínimo equivalente.

3. Modificación

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

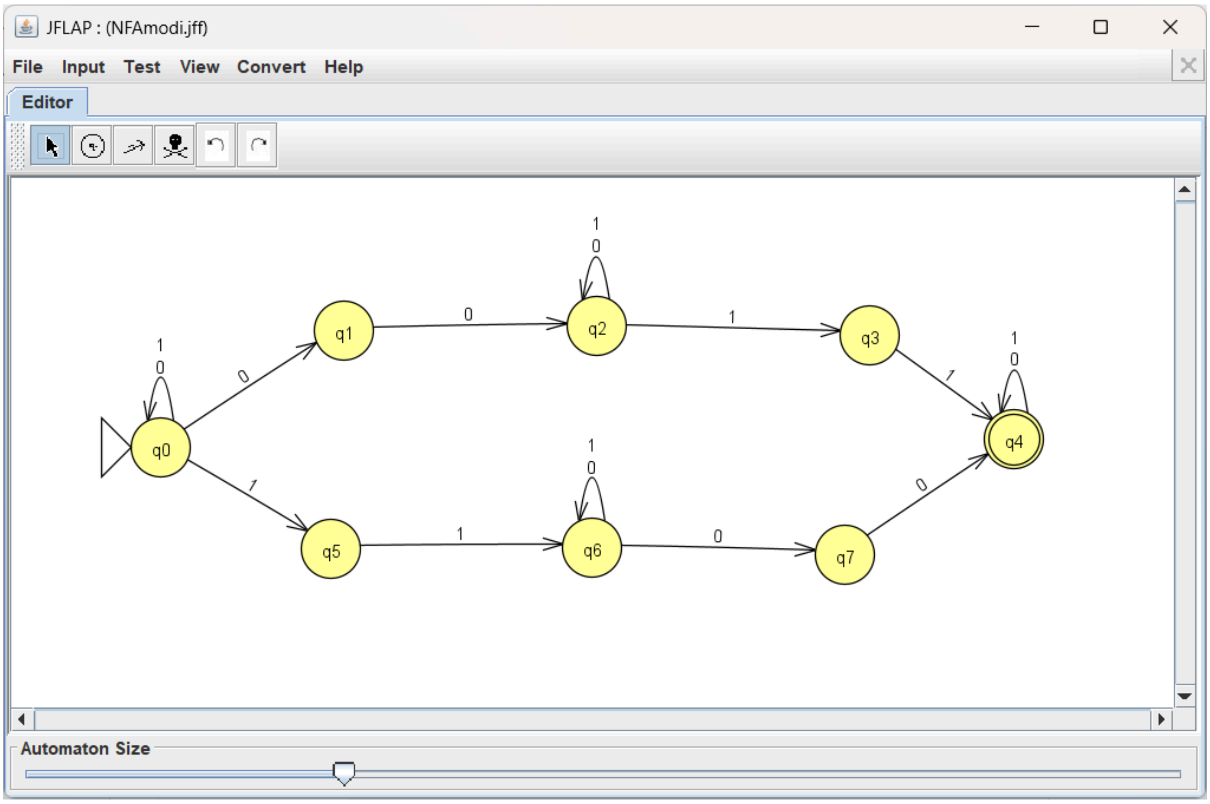

Figura 49: del lenguaje binario con dos ceros consecutivos y dos unos consecutivos.

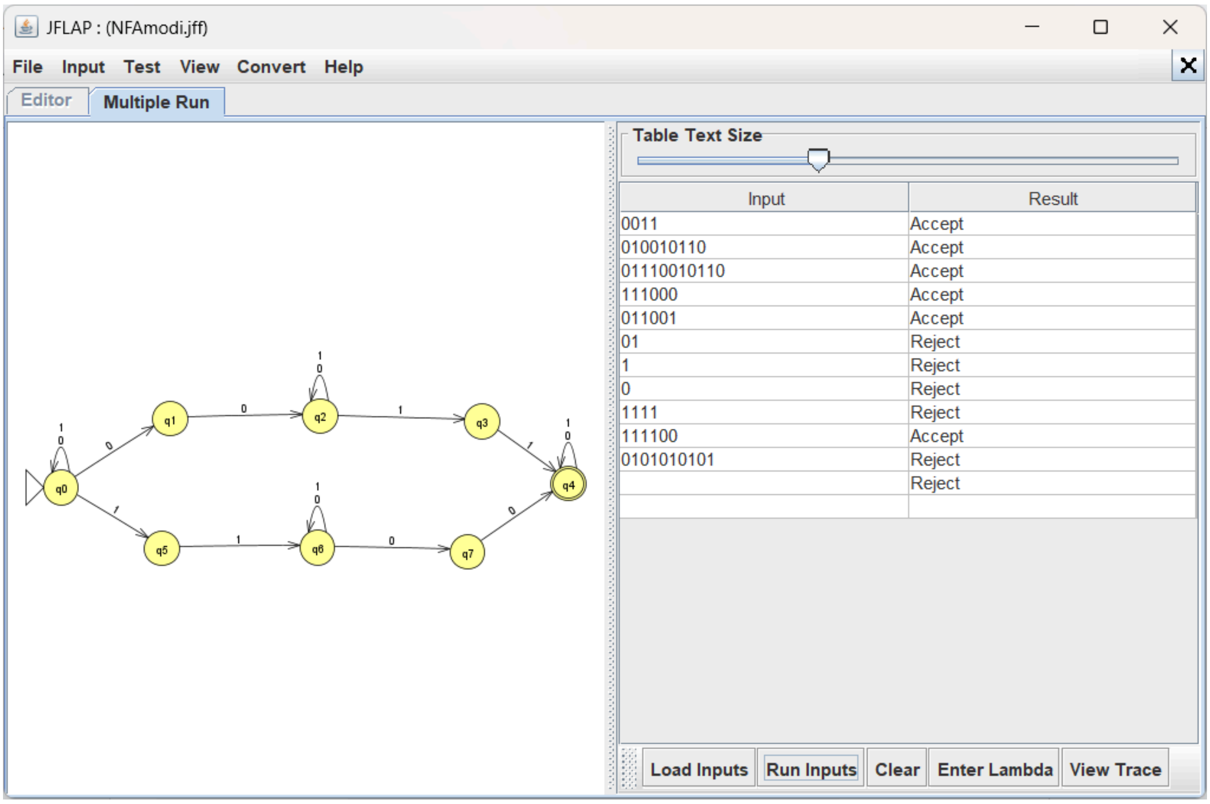

Figura 50: Cadenas de prueba para el NFA anterior.

Utilizamos el JFLAP para obtener un DFA equivalente:

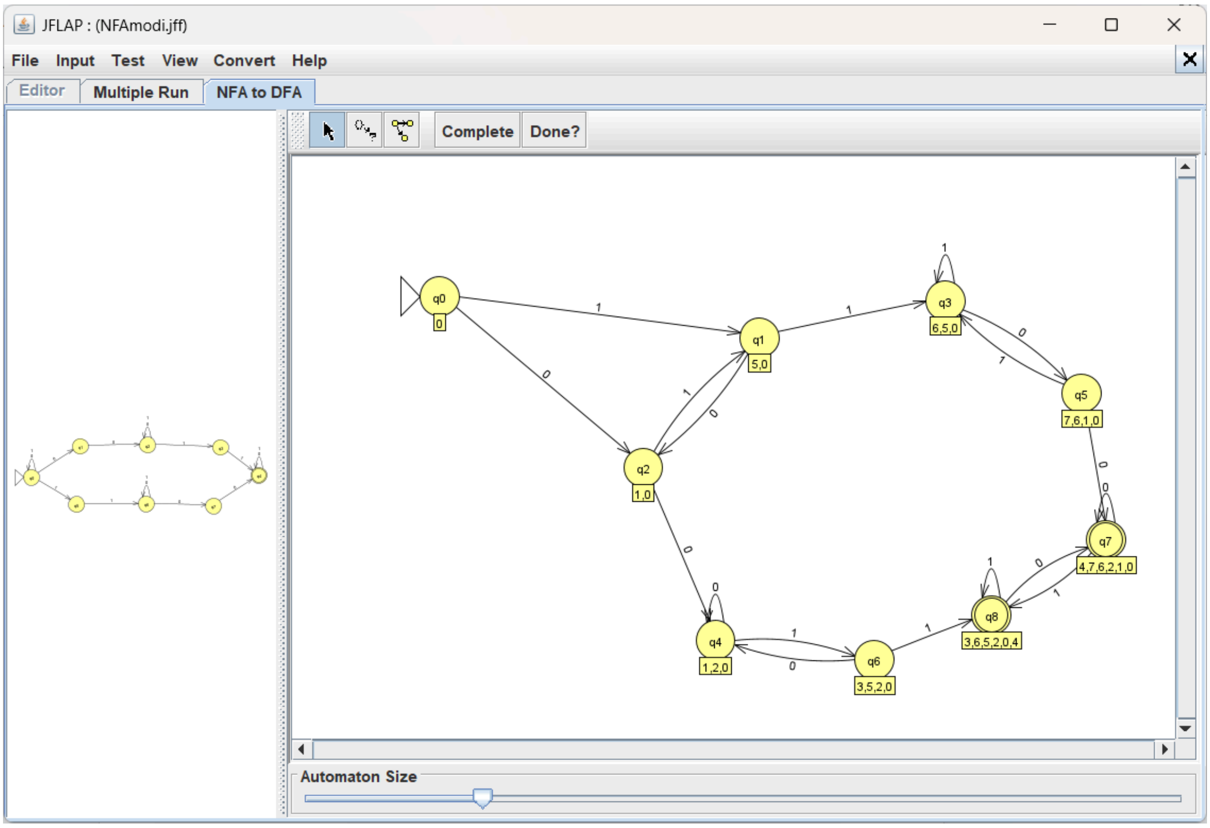

Figura 51: DFA equivalente al NFA anterior.

Utilizamos el JFLAP para minimizar el DFA anterior:

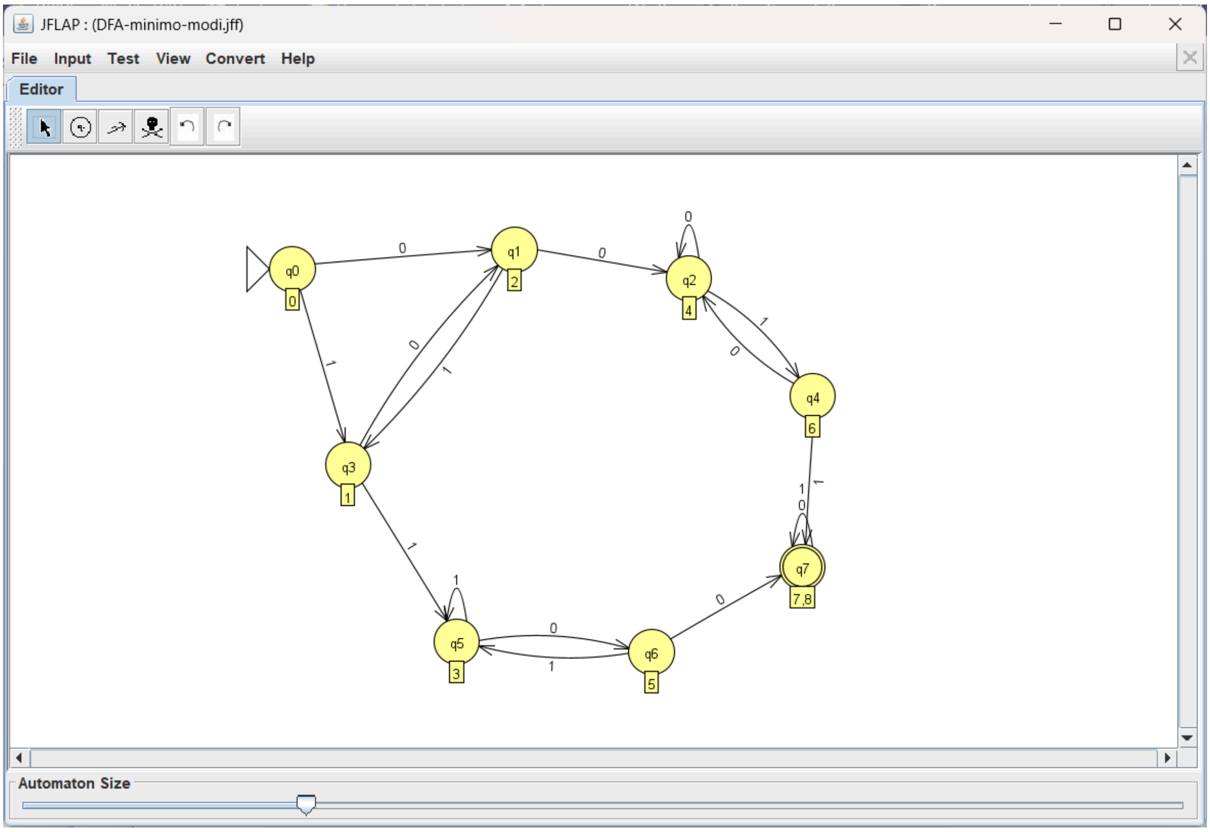

Figura 52: DFA mínimo equivalente.

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

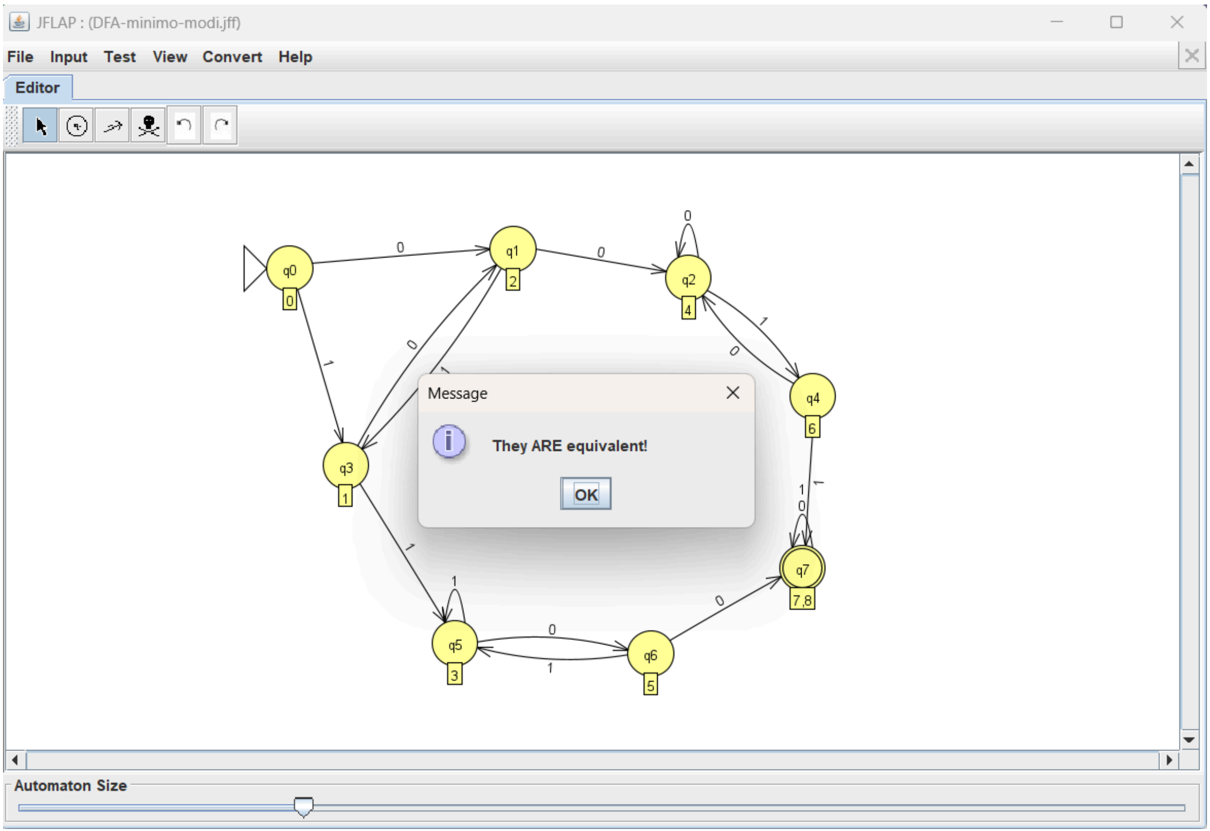

Figura 53: El DFA mínimo obtenido es equivalente al NFA del que partimos.