Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Informe Práctica 4

Autómatas finitos en JFLAP

Computabilidad y Algoritmia

Francesco Marelli
(alu0101161730@ull.edu.es)

Índice

Autómatas finitos en JFLAP

DFAs

Ejercicio 1

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

Σ = {a, b, c}

El autómata consta de dos estados: ***q0*** es el estado de arranque y ***q1*** el estado de aceptación. Con que la cadena del alfabeto ***Σ*** tenga longitud impar viene aceptada, por lo que con llegar al estado de aceptación será una cadena válida. Si nos fijamos en el número de transiciones para llegar a q1 siempre van a ser impares.

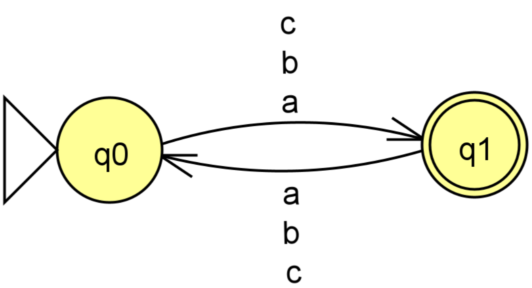


Table Text Size

Input	Result
aaa	Accept
abc	Accept
aaaaaaa	Accept
aabbc	Accept
ccc	Accept
	Reject
ab	Reject
abab	Reject
bc	Reject
aabbcc	Reject
bbbccc	Reject

Ejercicio 2

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

Σ = {a, b}

El autómata consta de cuatro estados: ***q0*** es el estado de arranque mientras que ***q1*** es el estado de aceptación. Se comprueba que todas las cadenas con un número de **a’s** par como **b’s** impar acaben en el estado de aceptación y las que no, no se reconozcan por el autómata.

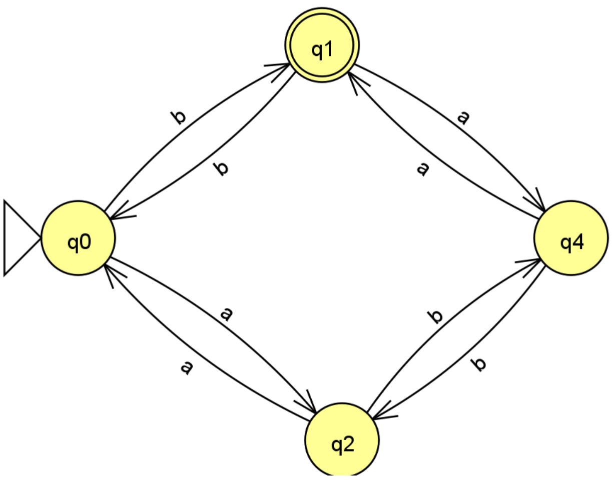


Table Text Size

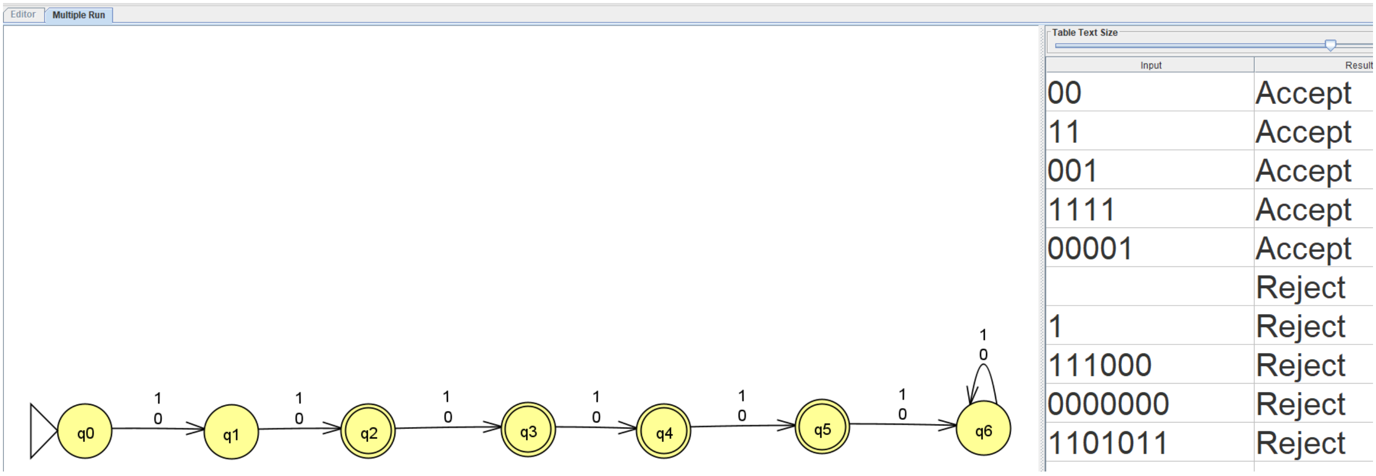
Input	Result
baa	Accept
ababbababbb	Accept
aab	Accept
bbb	Accept
ababb	Accept
abababa	Accept
abb	Reject
bba	Reject
bbbb	Reject
abbb	Reject
ab	Reject
	Reject

Ejercicio 3

Enunciado: Cadenas w sobre el alfabeto {0, 1} tales que 2 ≤ |w| ≤ 5.

Σ = {0,1}

El autómata consta de 7 estados. El estado de arranque es ***q0*** mientras que los estados de aceptación son ***q2, q3, q4, q5*** y ***q6*** representa un estado de muerte. Todas las cadenas del alfabeto que tengan longitud comprendida entre dos y cinco terminarán en un estado de aceptación, las que no, no serán aceptadas.

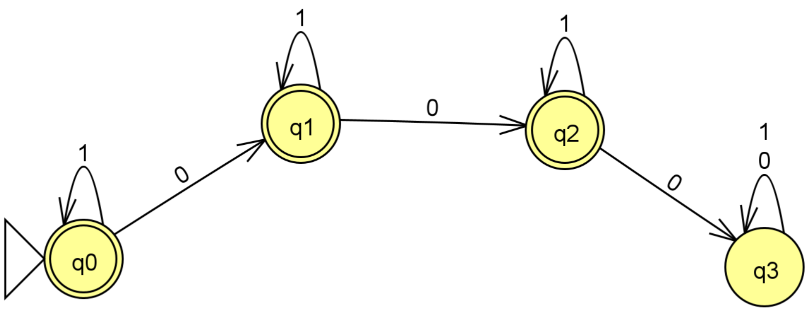


Ejercicio 4

Enunciado: Cadenas sobre el alfabeto {0, 1} que tengan como máximo dos ceros.

Σ = {0,1}

El autómata consta de cuatro estados, ***q0*** es el estado de arranque mientras que ***q0, q1, q2*** son los estados de aceptación. En el caso de ***q0*** es que también la cadena vacía será aceptada, puesto que lo único que el autómata comprueba es que no haya más de dos **0's** en ella**;** todas las cadenas que tengan como mucho esa cantidad de ***0's*** acabarán en un estado de aceptación.



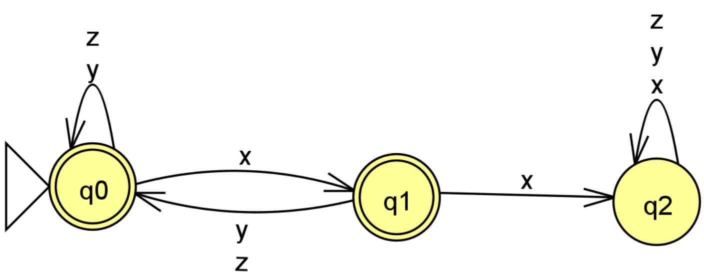
Input	Result
001111	Accept
1010111	Accept
	Accept
1010	Accept
1111101111011	Accept
1	Accept
0	Accept
000	Reject
1110000	Reject
101010	Reject
00110	Reject

Ejercicio 5

Enunciado: Cadenas sobre el alfabeto {x, y, z} que no contengan dos símbolos x consecutivos.

Σ = {x,y,z}

El autómata consta de tres estados, ***q0*** es el estado de arranque mientras que ***q0, q1*** son los estados de aceptación. En el caso de ***q0*** es que también la cadena vacía será aceptada, puesto que lo único que el autómata comprueba es que no haya más de dos **x's** consecutivas, cualquier otra cadena del alfabeto acabará en un estado de aceptación.



Input	Result
x	Accept
xyz	Accept
xyyyyyzzz	Accept
	Accept
xyxzyz	Accept
xx	Reject
yyyyxxx	Reject
yyxyxx	Reject
zzyyxx	Reject

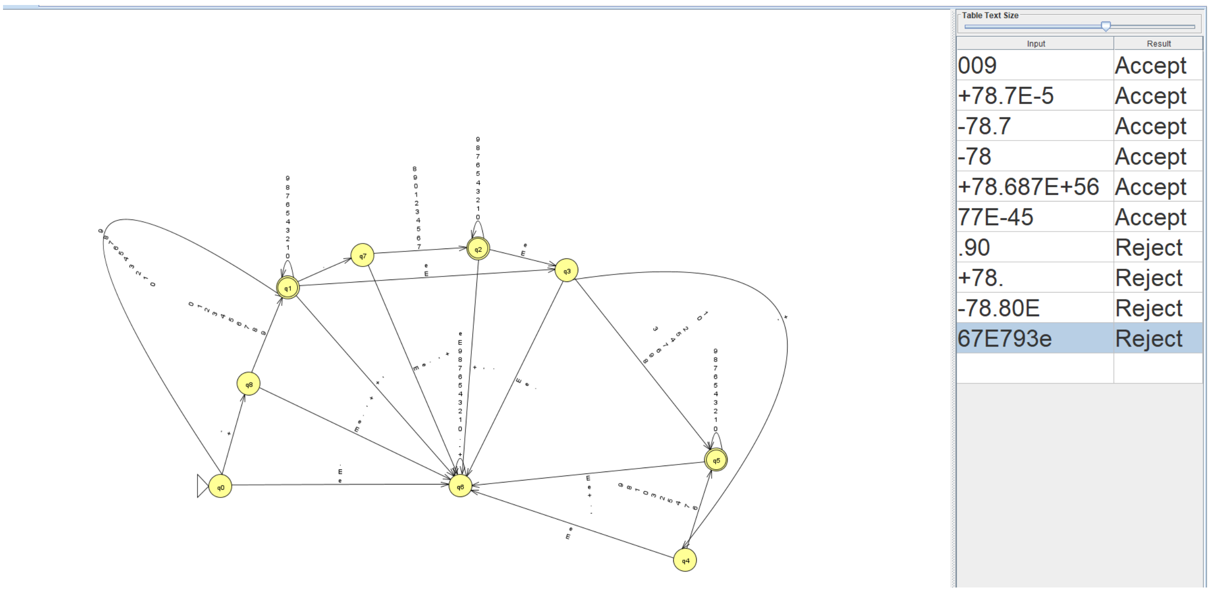
Ejercicio 6

Disẽnar un autómata finito determinista que acepte numeros reales. El alfabeto que usa el autómata se define como Σ = {+, −, ., E, e, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9} y las cadenas a aceptar se definen de la siguiente forma:

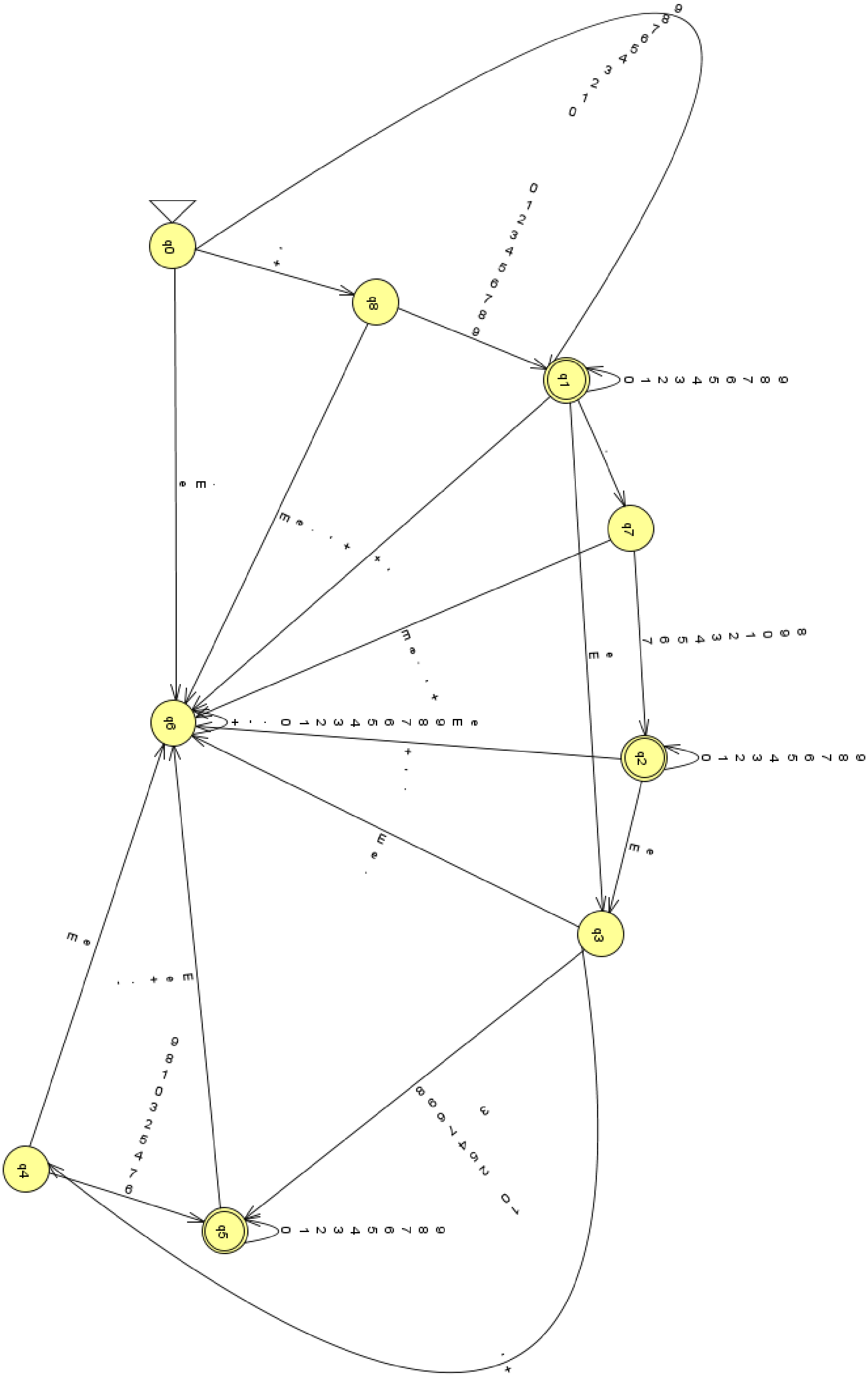
- La cadena comienza opcionalmente por el símbolo “+” o “-”
- A continuación la cadena contiene uno o varios símbolos en el rango [0 − 9]
- Posteriormente, y de forma opcional, aparece en la cadena el símbolo “.”. Si aparece este símbolo, la cadena debe continuar con uno o más símbolos entre el rango [0 − 9].
- Opcionalmente la cadena puede ir seguida del símbolo “E” o “e” para indicar un número en notación científica. En este caso, tendrá que ir seguido de un símbolo “+” o “-” (opcional) y una cadena de uno o más símbolos entre el rango [0-9] que representan el exponente.
- Algunas cadenas aceptadas por este autómata serán: 009, -78, -78.7, +78.7E-5, -7.876e+56, -78.87E56, etc.
- Algunas cadenas no aceptadas por este autómata serían: .90, +78., +78.90E, etc

Σ = {+, −, ., E, e, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

El autómata consta de diez estados, ***q0*** es el estado de arranque mientras que ***q1, q2 y q5*** son los estados de aceptación, ***q6*** es un estado de muerte. El autómata contempla todos los casos de número reales, tantos naturales como enteros o escritos con notación científica. Cualquiera de esas cadenas de números acabará en un estado de aceptación sino en el de muerte, por lo que vendrán rechazadas por el autómata.



Input	Result
009	Accept
+78.7E-5	Accept
-78.7	Accept
-78	Accept
+78.687E+56	Accept
77E-45	Accept
.90	Reject
+78.	Reject
-78.80E	Reject
67E793e	Reject


q0
q1
q2
q3
q4
q5
q6
q7
q8

NFAs

Ejercicio 1

Enunciado: Diseñar un autómata finito no determinista que reconozca cadenas sobre el alfabeto Σ = {a, b, c} que contengan la subcadena abc. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

Σ = {a, b, c}

El autómata consta de cuatro estados: ***q0*** es el de arranque y ***q3*** es el de aceptaci on. Se construyó el NFA considerando que cualquier secuencia de símbolos del alfabeto sería válida, antes y después de hallar la subcadena **abc**, acabando cualquier cadena que la contenga en el estado de aceptación.

NFA

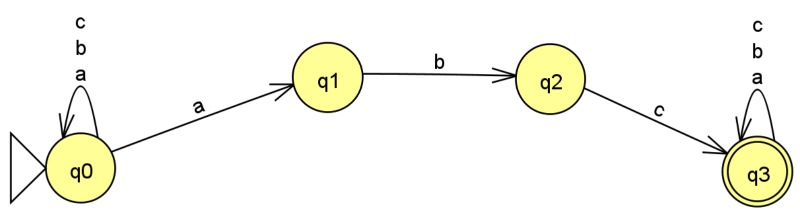


Table Text Size

Input	Result
abc	Accept
aaaaabcccc	Accept
bbbbbcccc...	Accept
abbbbbca...	Accept
abababab c	Accept
	Reject
aa	Reject
abbb	Reject
bcabb	Reject
cccc	Reject

DFA y DFA mínimo
Primero se convierte el NFA a DFA con el algoritmo de construcción de subconjuntos.

Apliquemos el algoritmo a mano;

NFA 1

$\varepsilon\text{-Clousure}(\{q_0\}) = \{q_0\} \equiv \boxed{A}$

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

$\varepsilon\text{-Clousure}(\{q_0,q_1\}) = \{q_0,q_1\} \equiv \boxed{B}$

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

$\delta(A,c) = \{q_0\} \equiv A$

$\delta(B,a) = \{q_0,q_1\} \equiv B$

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

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

$\delta(C,a) = \{q_0,q_1\} \equiv B$

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

$\delta(C,c) = \{q_0,q_3\} \equiv \boxed{D}$

$\delta(D,a) = \{q_0,q_1,q_3\} \equiv \boxed{E}$

$\delta(D,b) = \{q_0,q_3\} \equiv D$

$\delta(D,c) = \{q_0,q_3\} \equiv D$

$\delta(E,a) = \{q_0,q_1,q_3\} \equiv E$

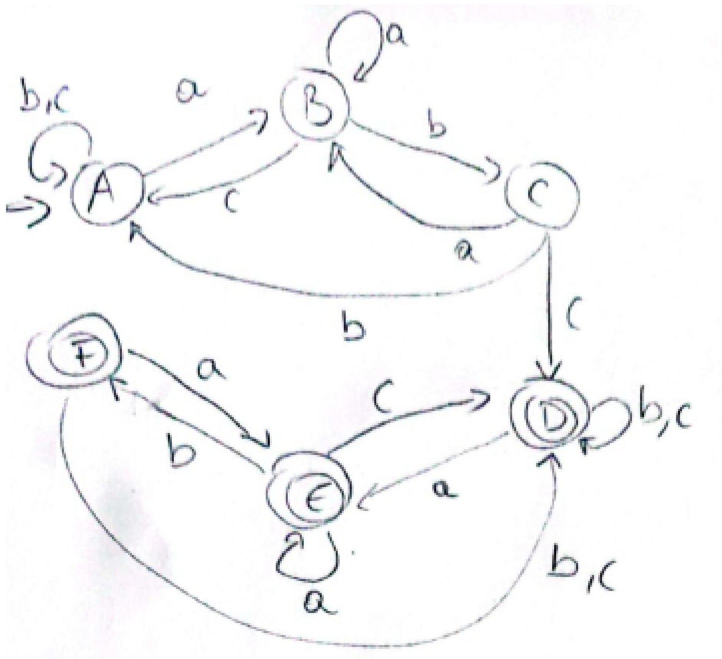
$\delta(E,b) = \{q_0,q_2,q_3\} \equiv \boxed{F}$

$\delta(E,c) = \{q_0,q_3\} \equiv D$

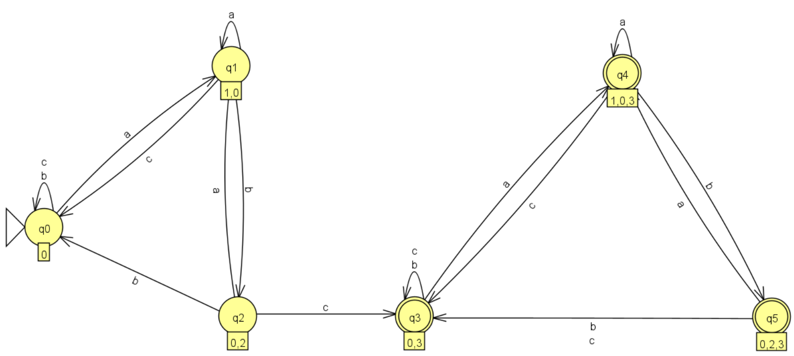
$\delta(F,a) = \{q_0,q_1,q_3\} = E$

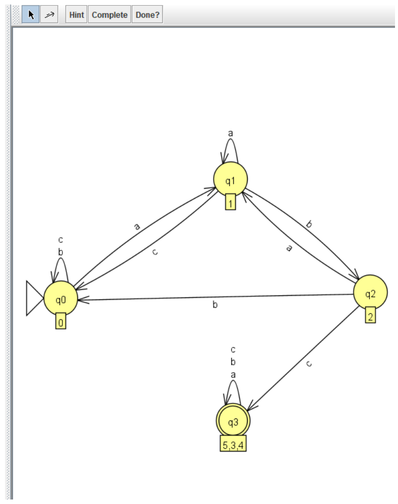
$\delta(F,b) = \{q_0,q_3\} = D$

$\delta(F,c) = \{q_0,q_3\} = D$



Ejecutemos en JFLAP la opción de crear un DFA.





Como se aprecia en las imágenes, el DFA calculado con JFLAP es equivalente al que se ha convertido con el algoritmo de construcción de subconjuntos. Después de eso se convierte el DFA obtenido a DFA mínimo, con el algoritmo de minimización de estados.

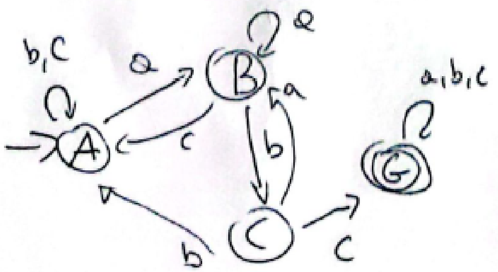
Apliquemos el algoritmo a mano:

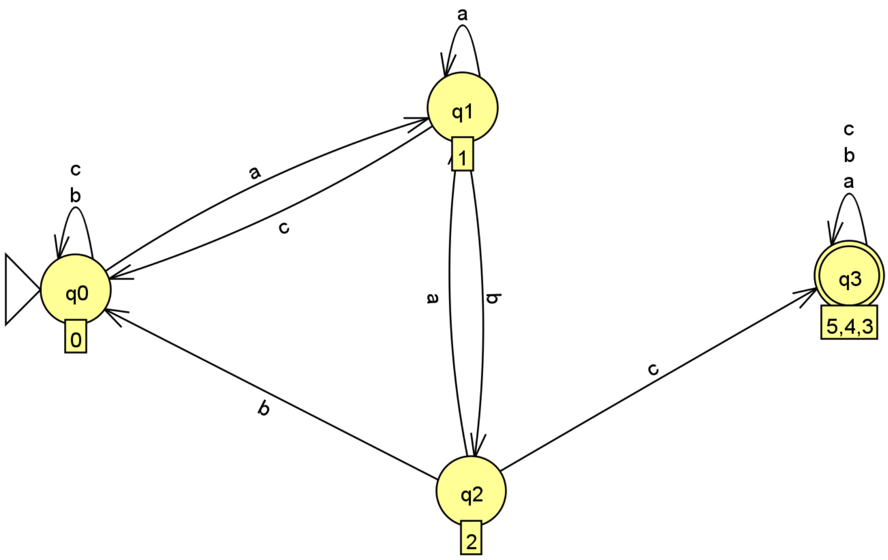
$\pi = \{D, E, F\} \{A, B, C\}$

$\{D, E, F\} \{C\} \{A, B\}$

$\{D, E, F\} \{C\} \{A\} \{B\}$

G Ahora es único estado de aceptación





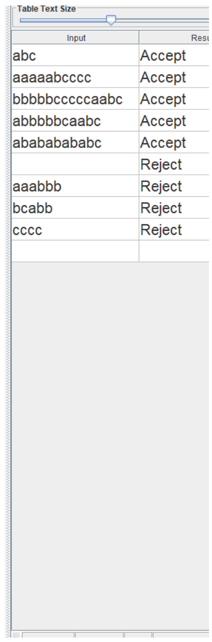


Table Text Size

Input	Resu
abc	Accept
aaaaabcccc	Accept
bbbbbcccccaabc	Accept
abbbbbcaabc	Accept
abababababc	Accept
	Reject
aaabbb	Reject
bcabb	Reject
cccc	Reject

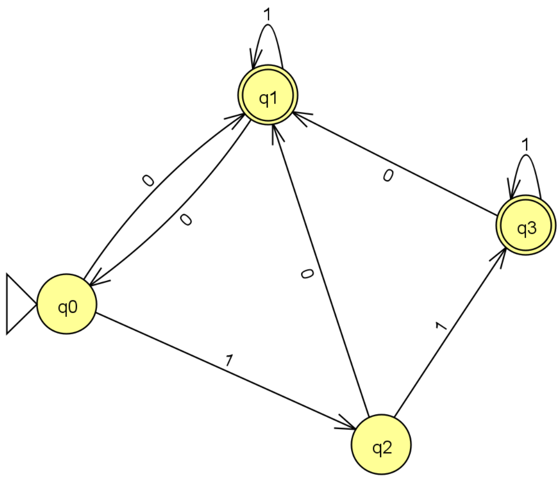
Como es apreciable en la figura y en la imagen , el DFA mínimo es el mismo que se ha resuelto a lápiz y papel y ambos reconocen las mismas cadenas del NFA que se ha convertido, por lo que son equivalentes.

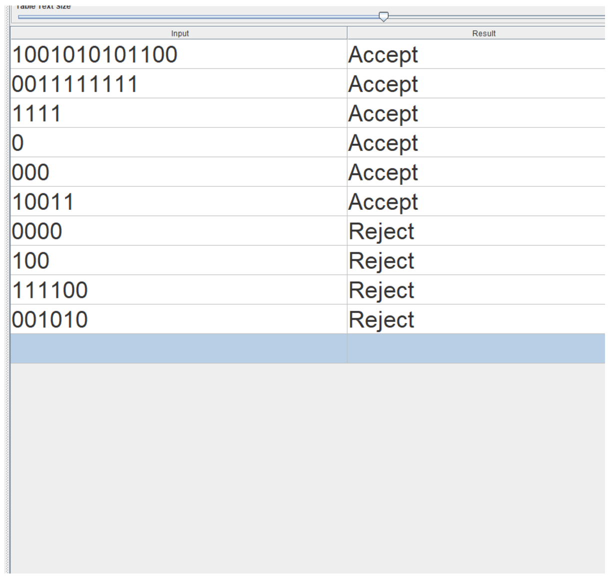
Ejercicio 2

Enunciado: Disẽnar un autómata finito no determinista que reconozca cadenas sobre el alfabeto Σ = {0, 1} tales que tienen un numero impar de ceros o terminan en 11. A partir del NFA diseñado, obtenga un DFA mínimo equivalente.

Σ = {0, 1}

El autómata consta de cuatro estados, ***q0*** es el estado de arranque y ***q1, q3*** son los estados de aceptación.. La rama de arriba del autómata se ocupa de comprobar la disparidad de los número de ***0's*** en las cadenas que se aceptan y en la parte inferior se comprueba que en caso contrario las cadenas aceptadas tengan que acabar por ***11.***

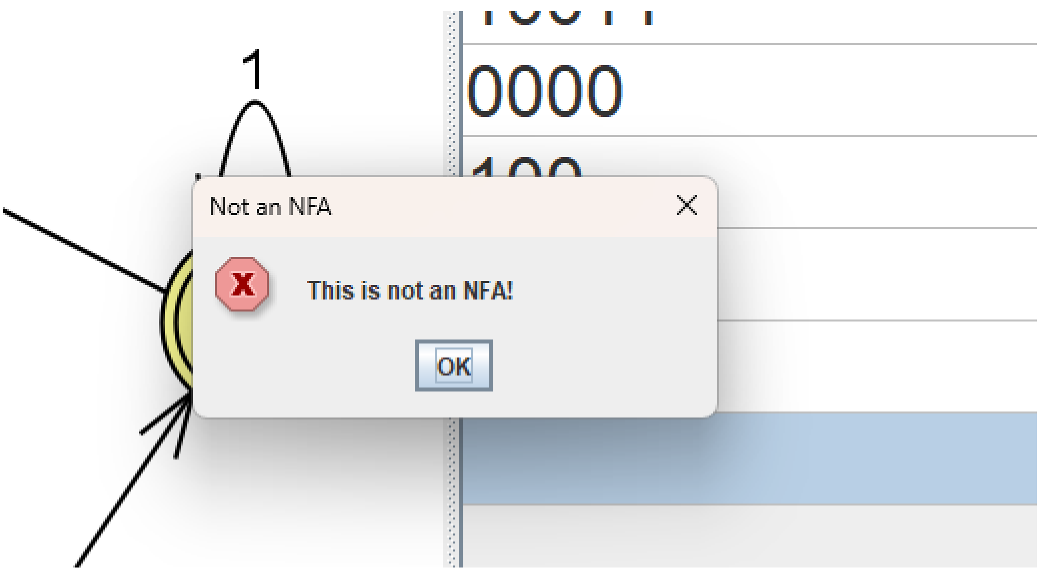




Input	Result
1001010101100	Accept
0011111111	Accept
1111	Accept
0	Accept
000	Accept
10011	Accept
0000	Reject
100	Reject
111100	Reject
001010	Reject

DFA y DFA mínimo

El autómata que se ha construído ya es un DFA, por qué tiene todos los estados ya definidos para cada símbolo. En un DFA, la función de transición devuelve un conjunto de un solo estado para cada símbolo del alfabeto. Esto significa que, para cada símbolo, el conjunto calculado siempre será un conjunto de un solo estado. Por lo tanto, el algoritmo de construcción por subconjuntos simplemente agrega un estado nuevo al conjunto de estados para cada símbolo del alfabeto. Como resultado, el DFA resultante será idéntico al NFA original.



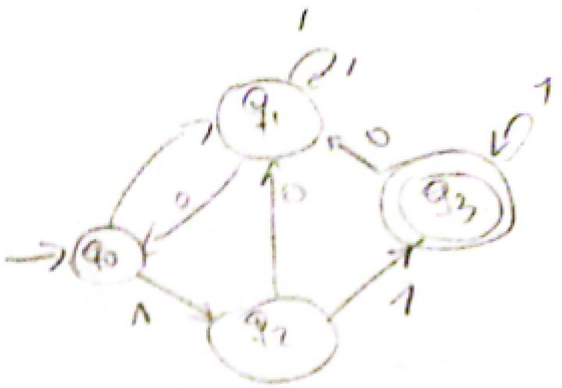
Después de eso, se minimiza el DFA obtenido a DFA mínimo, con el algoritmo de minimización de estados.

Apliquemos el algoritmo:

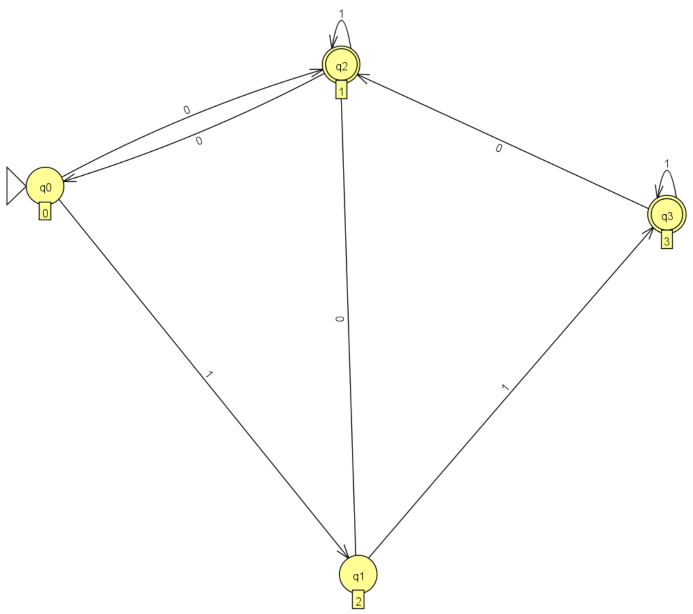
$$\pi = \{q_1, q_3\} \; \{q_0, q_2\}$$

$$\{q_1, q_3\} \; \{q_0\} \; \{q_2\}$$

$$\{q_1\} \; \{q_3\} \; \{q_0\} \; \{q_2\}$$



El resultado es el mismo DFA que ya se intuía ser mínimo, y el mismo que produjo JFLAP. Ambos DFAs reconocen las mismas cadenas.



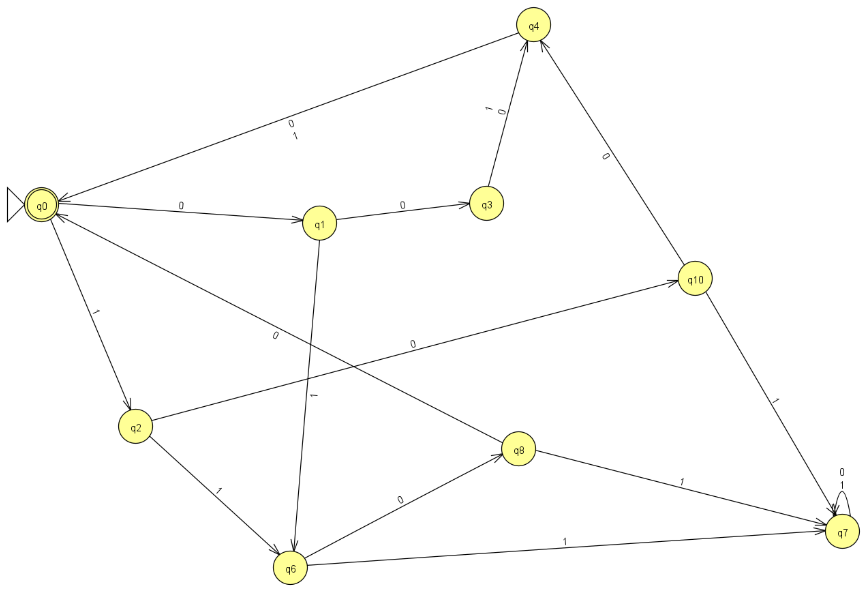
Input	Result
1001010101100	Accept
0011111111	Accept
1111	Accept
0	Accept
000	Accept
10011	Accept
0000	Reject
100	Reject
111100	Reject
001010	Reject

Ejercicio 3

Enunciado: Diseñar un autómata finito no determinista que reconozca todas las cadenas binarias w, con |w| % 4 = 0, en las que cada bloque de cuatro símbolos contiene al menos dos ceros consecutivos. Los bloques comenzarán siempre en posiciones múltiplo de 4.

Σ = {0, 1}

El autómata consta de nueve estados, ***q0*** es el estado de arranque y aceptación. . Lo que el autómata reconoce son cadenas no solo cuya longitud sea |4| sino que es las subcadenas o bloque de longitud |4| esté la subcadena ***00*** es decir que cada 4 posiciones tiene que haber una secuencia de ceros. No se aceptará una cadena que tenga dos ceros seguidos pero en posiciones de bloques diferentes.

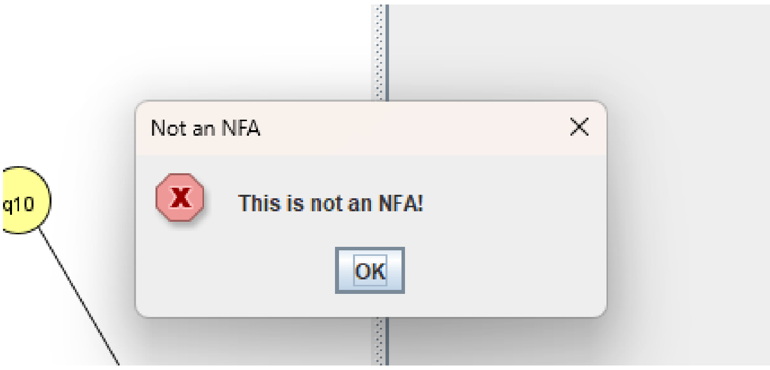


Input	Result
00111100	Accept
	Accept
11000001	Accept
0000	Accept
110011001001	Accept
11100111	Reject
000	Reject
00001111	Reject
10011001101	Reject

DFA y DFA mínimo

Primero se convierte el NFA a DFA con el algoritmo de construcción de subconjuntos. El autómata que se ha construído ya es un DFA, por qué tiene todos los estados ya definidos para cada símbolo. En un DFA, la función de transición devuelve un conjunto de un solo estado para cada símbolo del alfabeto. Esto significa que, para cada símbolo, el conjunto calculado siempre será un conjunto de un solo estado. Por lo tanto, el algoritmo de construcción por

subconjuntos simplemente agrega un estado nuevo al conjunto de estados para cada símbolo del alfabeto.
Como resultado, el DFA resultante será idéntico al NFA original.

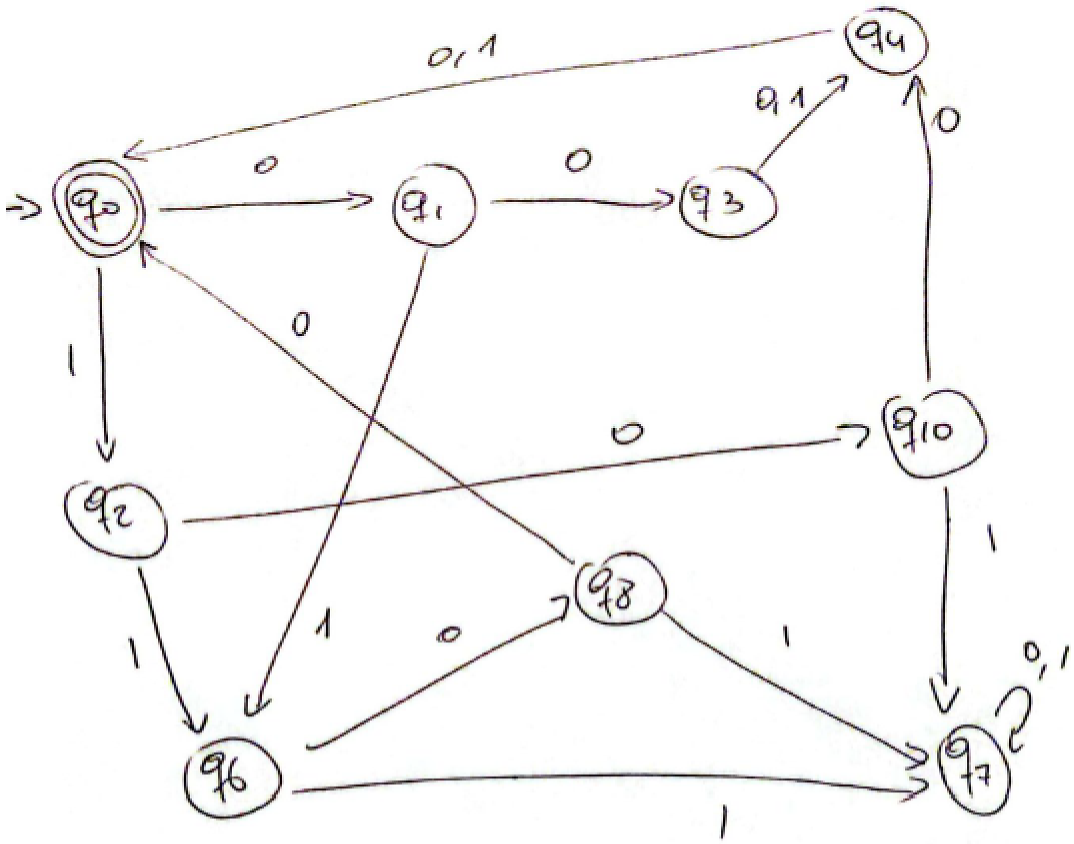


Después el DFA obtenido se pasa DFA mínimo, con el algoritmo de minimización de estados.

Apliquemos el algoritmo:

$$\Pi = \{q_0\}\ \{q_1, q_3, q_2, q_6, q_8, q_7, q_{10}, q_4\}$$
$$\{q_0\}\ \{q_8\}\ \{q_1, q_3, q_2, q_6, q_7, q_{10}, q_4\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1, q_3, q_2, q_7, q_{10}, q_4\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1\}\ \{q_3, q_2, q_7, q_{10}, q_4\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1\}\ \{q_2\}\ \{q_3, q_7, q_{10}, q_4\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1\}\ \{q_2\}\ \{q_4\}\ \{q_3, q_7, q_{10}\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1\}\ \{q_2\}\ \{q_4\}\ \{q_7\}\ \{q_3, q_{10}\}$$
$$\{q_0\}\ \{q_8\}\ \{q_6\}\ \{q_1\}\ \{q_2\}\ \{q_4\}\ \{q_7\}\ \{q_3\}\ \{q_{10}\}$$

El resultado es el mismo DFA que ya era mínimo



El DFA mínimo resultante reconoce a las mismas cadenas del autómata de partida desde que lo construimos.