Ingeniería en Sistemas de Información

Sintaxis y Semántica del Lenguaje

RESOLUCIÓN PRÁCTICO N2

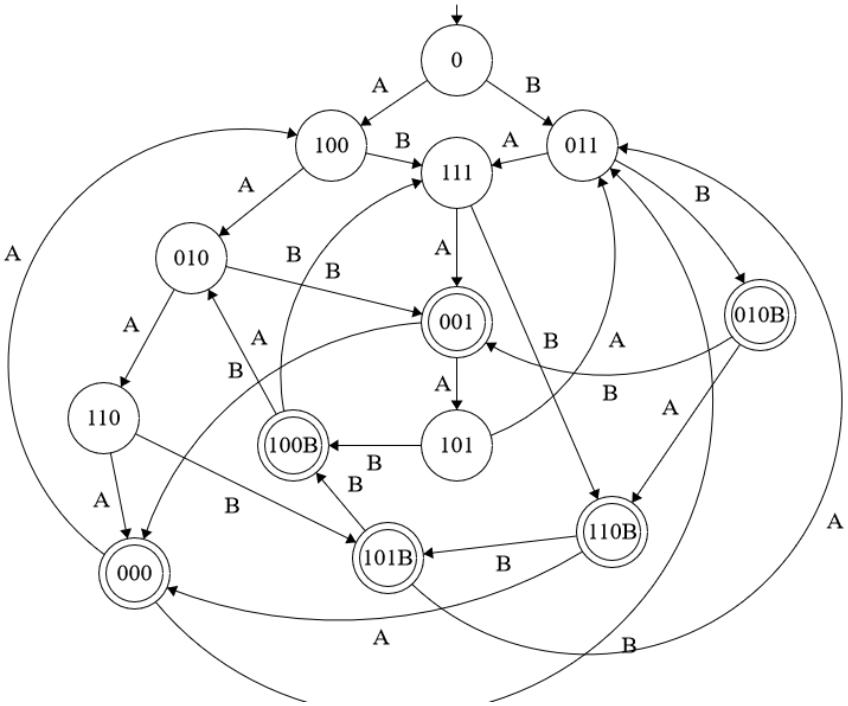
"Diagramas de Transición de Estados – Autómatas Finitos – Lenguajes Regulares".

Profesores:

Ing. Mario Rinaldi
Dr. Jorge A. Palombarini

29 DE SEPTIEMBRE DE 2020
UNIVERSIDAD TECNOLÓGICA NACIONAL FRVM
Mauricio Achetta – Ignacio Guridi – Natanael Bertello.

1) A)



Parte 2:

a-

```
from automata.base.automaton import Automaton
from automata.fa.dfa import DFA
dfa = DFA(
    states=('0','100','011','010','111','010B','110','001','110B','101','100B','000'),
    input_symbols=('A', 'B'),
    transitions={
        '0': {'A':'100', 'B':'011'},
        '100': {'A':'010', 'B':'111'},
        '011': {'A':'111', 'B':'010B'},
        '010': {'A':'110', 'B':'001'},
        '111': {'A':'001', 'B':'110B'},
        '010B': {'A':'110B', 'B':'001'},
        '110': {'A':'000', 'B':'101B'},
        '001': {'A':'101', 'B':'000'},
        '110B': {'A':'000', 'B':'101B'},
        '101': {'A':'011', 'B':'100B'},
        '100B': {'A':'010', 'B':'111'},
        '101B': {'A': '011', 'B': '100B'},
        '000': {'A': '100', 'B': '011'},

    },
    initial_state='0',
    final_states={'001','010B','100B','110B','101B','000'}
)
```

```
dfa.validate()
```

```
True

Process finished with exit code 0
```

b-

```
ejemplo = ["BBB", "ABA", "AAAA", "BB", "BBAB", "BAAAB", "AABAB", "BBA", "BAB", "AAAB", "AAAAB", "AAAAB", "BAAA", "ABAA", "AB", "BA", "AAA", "B", "A"]
cadenaaceptada = 0
cadenarechazada = 0
for x in range(0, len(ejemplo)):
    if dfa.accepts_input(ejemplo[x]):
        cadenaaceptada += 1
        print("La cadena", ejemplo[x]+ " fue aceptada porqué el automata se detiene en el estado ", dfa.read_input(ejemplo[x]))
    else:
        cadenarechazada += 1
        print("La cadena", ejemplo[x]+ " fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada")
```

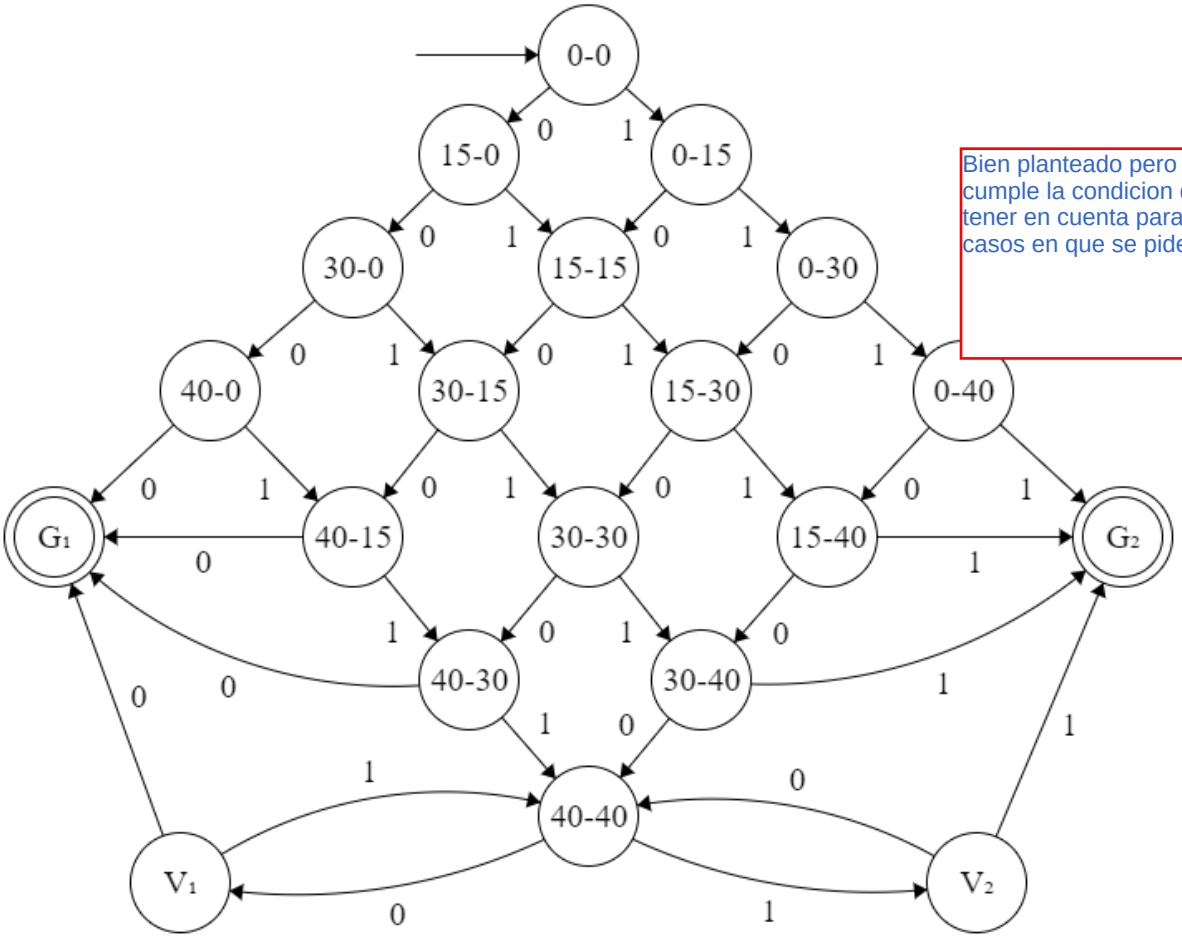
```
La cadena BBB fue aceptada porqué el automata se detiene en el estado  001
La cadena ABA fue aceptada porqué el automata se detiene en el estado  001
La cadena AAAA fue aceptada porqué el automata se detiene en el estado  000
La cadena BB fue aceptada porqué el automata se detiene en el estado  010B
La cadena BBAB fue aceptada porqué el automata se detiene en el estado  101B
La cadena BAAAB fue aceptada porqué el automata se detiene en el estado  100B
La cadena AABAB fue aceptada porqué el automata se detiene en el estado  100B
La cadena BBA fue aceptada porqué el automata se detiene en el estado  110B
La cadena BAB fue aceptada porqué el automata se detiene en el estado  110B
La cadena AAAB fue aceptada porqué el automata se detiene en el estado  101B
La cadena AAAAB fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena AAAAB fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena BAAA fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena ABAA fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena AB fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena BA fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena AAA fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena B fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena A fue rechazada porqué el automata se deteine en un estado de no aceptacion, por lo tanto la cadena es denegada

Process finished with exit code 0
```

c-

```
minimal_dfa = dfa.minify()
if minimal_dfa == dfa:
    print("El automata utilizado es el mínimo posible")
else:
    print("El automata utilizado no es el mínimo posible")
```

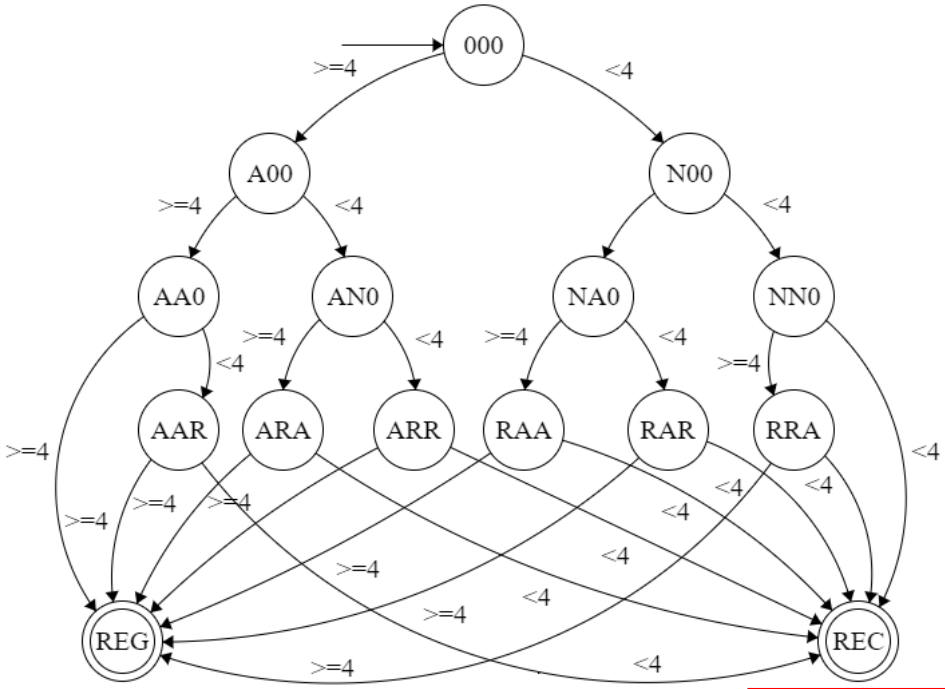
```
El automata utilizado es el mínimo posible

Process finished with exit code 0
```

B)



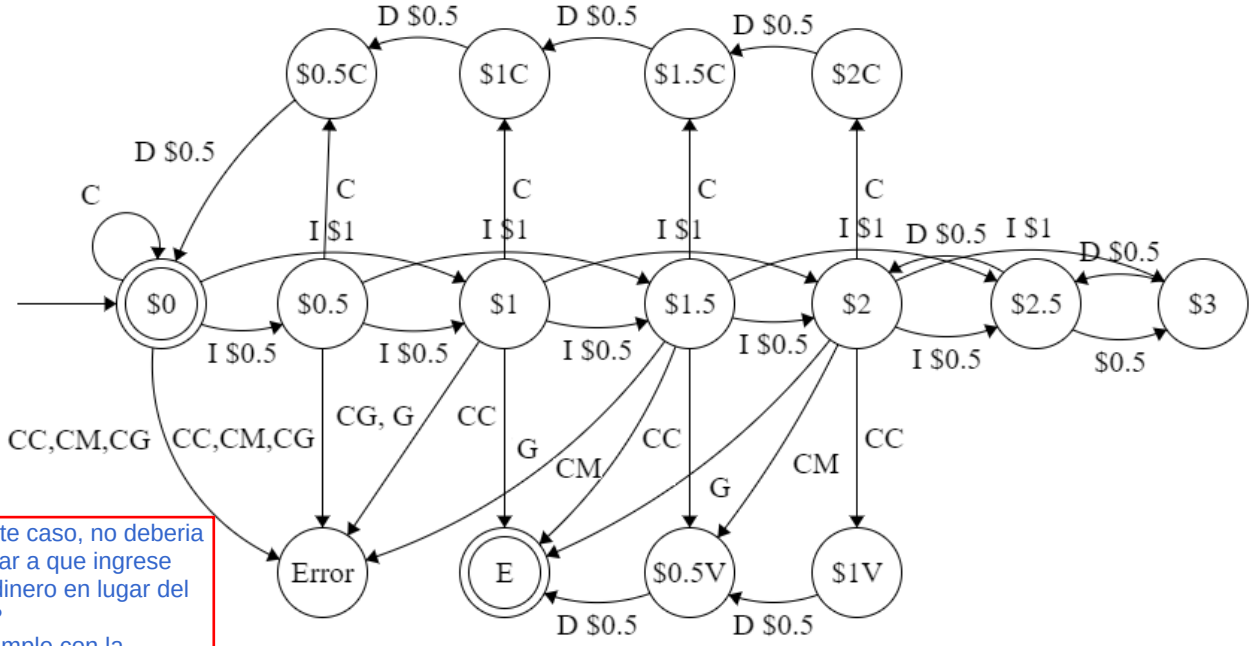
Bien planteado pero no cumple la condicion de AFD, tener en cuenta para los casos en que se pide AFD

C) * 0: El parcial aún no ha sido tomado.
* A: Parcial aprobado.
* N: Parcial no aprobado.
* R: Recuperatorio.



No cumple con la condicion de AFD, idem anterior

D) * I: Ingresar dinero.
* D: Devolver dinero.
* C: Cancelar.
* CC: Café Chico.
* CM: Café Mediano.
* G: Gaseosa.
* V: Valor a devolver.



En este caso, no deberia esperar a que ingrese mas dinero en lugar del error?
No cumple con la condicion de AFD

2) **AUTOMATA 1:**

A. A := $(Q; \sum; q0; F; \delta)$ $Q = \{q1, q2, q3\}$ $\sum = \{0,1\}$ $q0 = q1$ $F = \{q2\}$

δ	0	1

q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

B. El lenguaje regular que reconoce el Autómata es: L = {w|w contiene al menos un **1**, y termina en **1** o un número par de **0**}.
Ejemplos: 1, 01, 11, 001, 011, 100, 101, 111, etc.

AUTÓMATA 2:

A. A := (Q; ∑; q0; F; δ) $Q = \{s, q1, q2, r1, r2\}$ $\sum = \{a, b\}$ $q0 = s$ $F = \{q1, r1\}$

δ	a	b
s	q_1	r_1
q_1	q_1	q_2
q_2	q_1	q_2
r_1	r_2	r_1
r_2	r_2	r_1

B. El lenguaje regular que reconoce el Autómata es: L = {w|w comienza y termina con **a**, o comienza y termina con **b**}.
Ejemplos: a, b, aa, bb, aaa, aba, bab, bbb, etc.

AUTÓMATA 3:

A. A := (Q; ∑; q0; F; δ) $Q = \{q0, q1, q2\}$ $\sum = \{0,1,2\}$ $q0 = q0$ $F = \{q0\}$

δ	0	1	2
q_0	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

B. El lenguaje regular que reconoce el Autómata es: L = {w|w es la cadena vacía, o está formada por **0**, o contiene igual cantidad de **1** y **2**, o solo está formada por uno de ellos cuya cantidad es divisible por tres}. Y que ocurre cuando aparece reset?
Ejemplos: λ, 0, 00, 12, 21, 000, 012, 021, 102, 111, 120, 201, 210, 222, etc.

AUTÓMATA 4:

A. A := (Q; ∑; q0; F; δ) $Q = \{q0, q1\}$ $\sum = \{letra, digito\}$ $q0 = q0$ $F = \{q1\}$

δ	letra	dígito
q_0	q_1	q_0
q_1	q_1	q_1

B. El lenguaje regular que reconoce el Autómata es: L = {w|w contiene al menos una vez el símbolo **letra**}.
Ejemplos: letra, dígitoletra, letraletra, letradígito, dígitodígitoletra, etc.

AUTÓMATA 5:

A. A := (Q; ∑; q0; F; δ) $Q = \{q0, q1, q2, q3\}$ $\sum = \{a, b\}$ $q0 = q0$ $F = \{q2\}$

δ	a	b
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_2
q_3	q_3	q_3

B. El lenguaje regular que reconoce el Autómata es: L = {w|w contiene exactamente dos **a**}.
Ejemplos: aa, aba, baa, aabb, abab, abba, baab, baba, bbaa, etc.

AUTÓMATA 6:

A. A := (Q; ∑; q0; F; δ) $Q = \{q0, q1, q2\}$ $\sum = \{a, b\}$ $q0 = q0$ $F = \{q0\}$

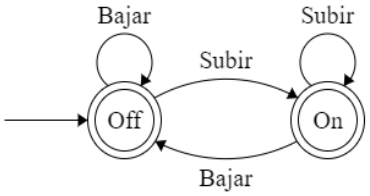
δ	a	b
q_0	q_1	q_2
q_1	q_2	q_0
q_2	q_2	q_2

B. El lenguaje regular que reconoce el Autómata es: L = {w|w es la cadena vacía o está formada por cualquier cantidad de veces la subcadena **ab** (No puede tener dos a o dos b consecutivas)}.
Ejemplos: λ, ab, abab, ababab, abababab, etc.

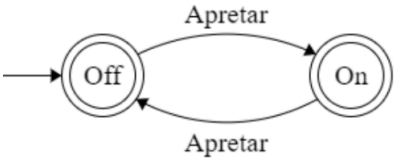
3) El autómata reconoce las cadenas que cumplen al menos una de las siguientes condiciones: es la cadena vacía; están formadas por **0**; contienen igual cantidad de **1** y **2**; solo están formadas por uno de ellos cuya cantidad es divisible por tres; o están formadas por subcadenas que cumplen las condiciones anteriores.

En el caso de que haya RESET no se cumple

4) Si el switch on/off que consiste en un interruptor que hay que cambiar de posición subiéndolo o bajándolo:
A := (Q; ∑; q0; F; δ) $Q = \{Off, On\}$ $\sum = \{\text{Bajar}, \text{Subir}\}$ $q0 = Off$ $F = \{Off, On\}$

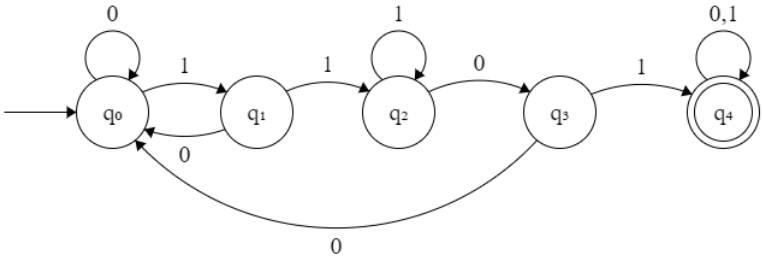


En cambio, si se trata de un switch on/off que consiste en un botón:
A := (Q; ∑; q0; F; δ) $Q = \{Off, On\}$ $\sum = \{\text{Apretar}\}$ $q0 = Off$ $F = \{Off, On\}$



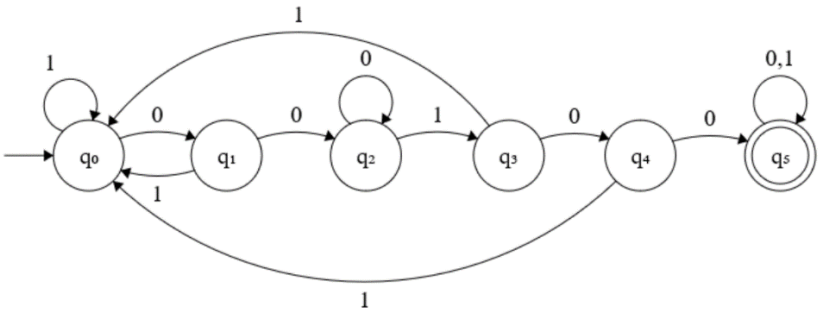
5) A)

Incompletos, falta la definicion formal y tambien el codigo en python.

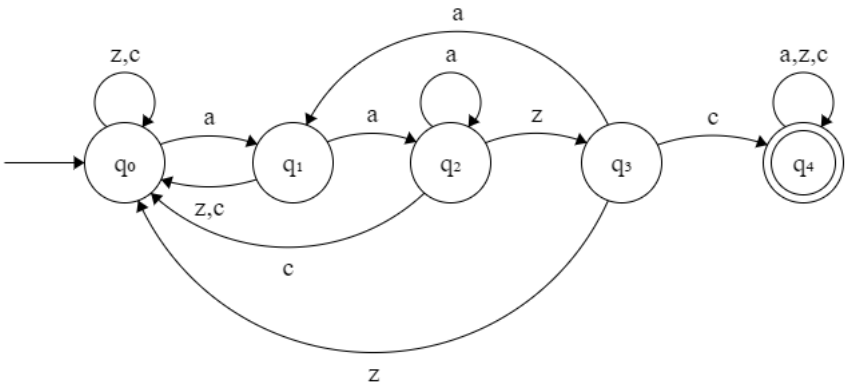


B)

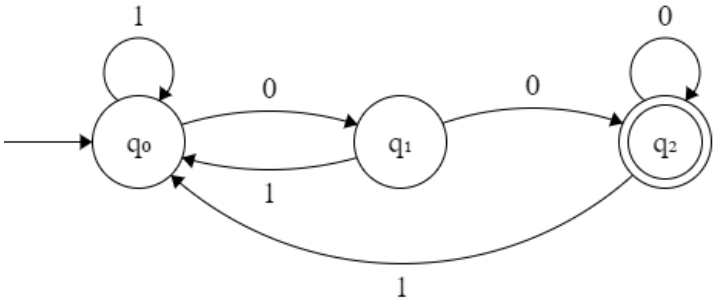
C)



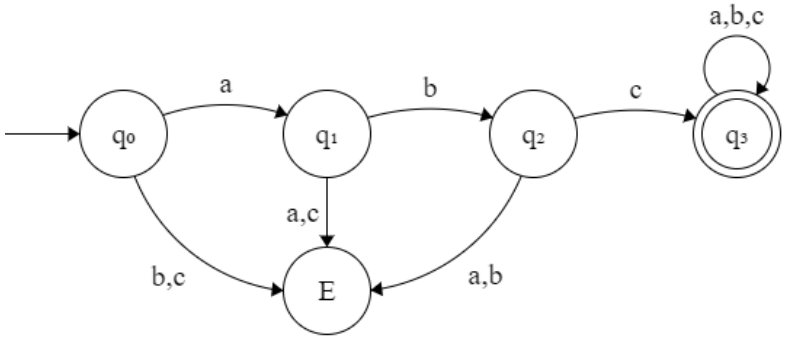
D)



E)

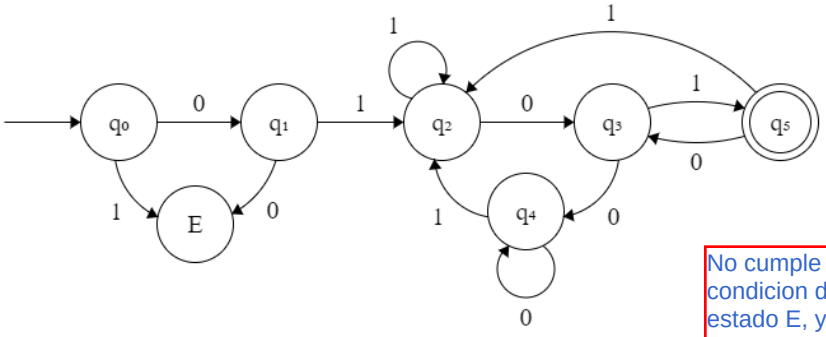


F)



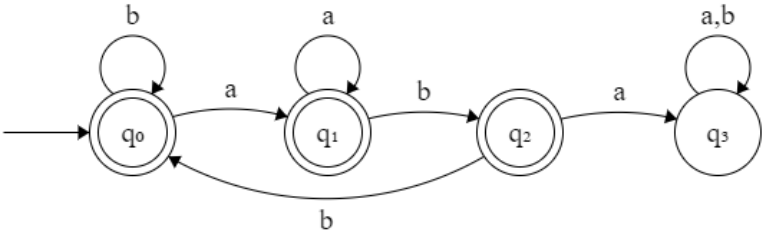
No cumple con la condicion de AFD en el estado E, ya que faltan transiciones

G)

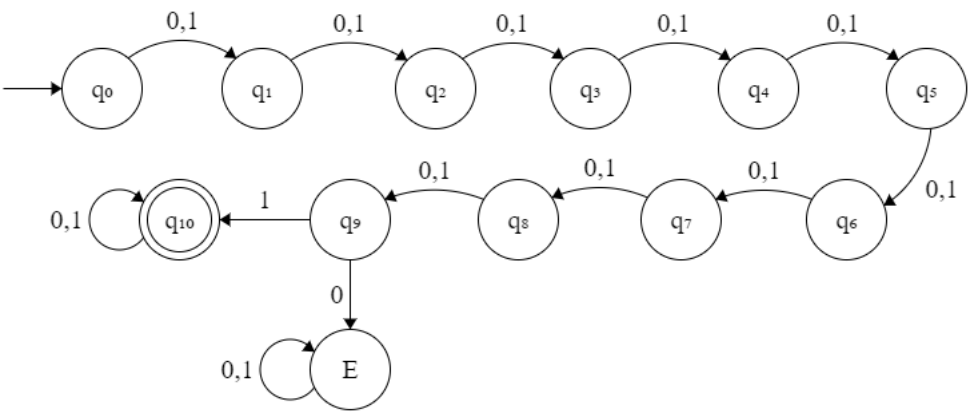


No cumple con la condicion de AFD en el estado E, ya que faltan transiciones

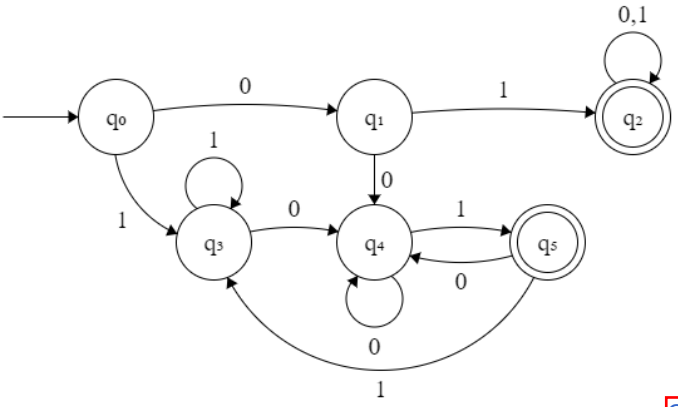
H)



I)

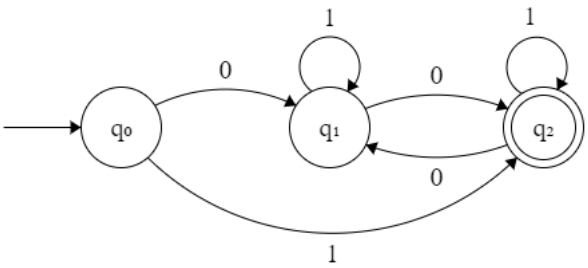


J)

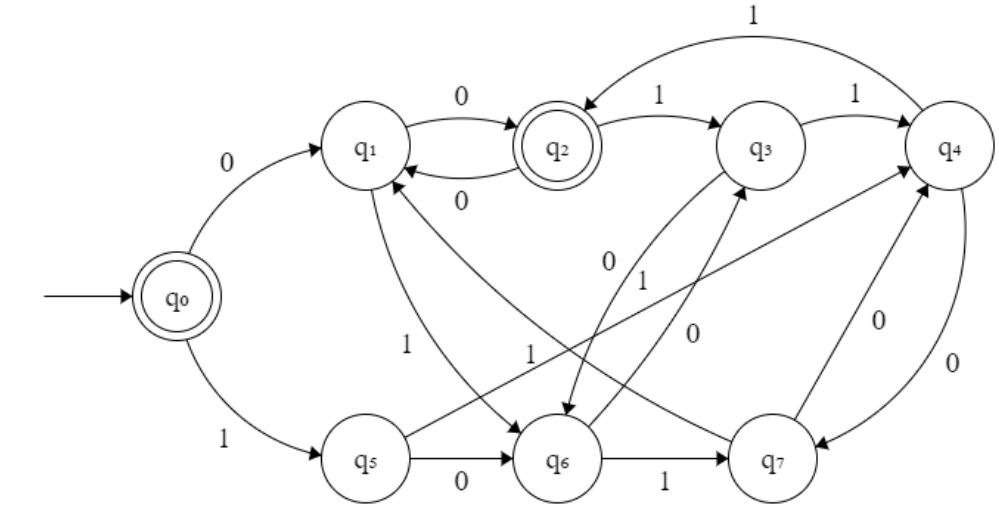


Cantidad 0 se toma como par y aqui no.

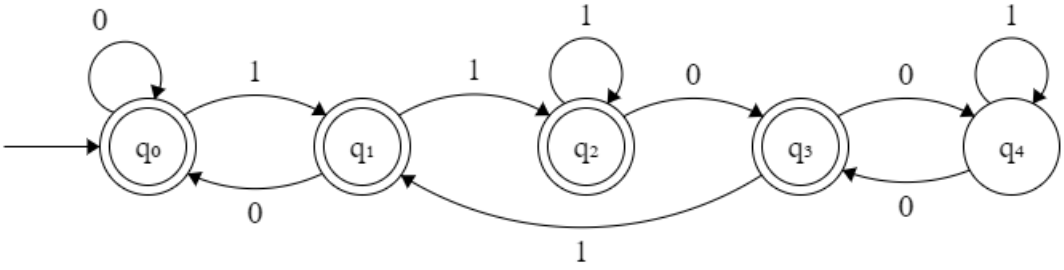
K)



L)

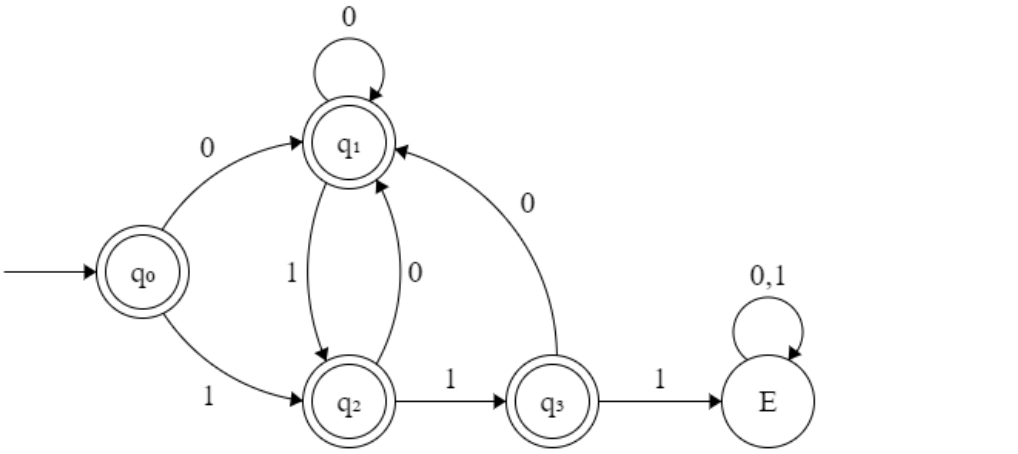


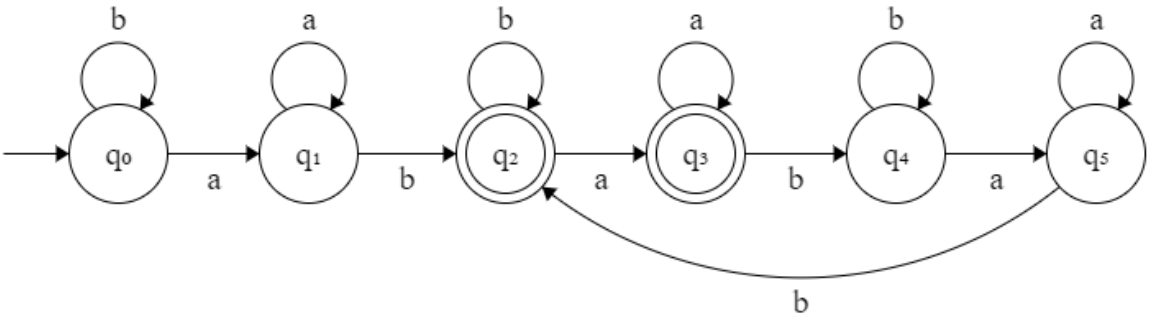
M)



Acepta cadenas no validas como 110100 y no deberia

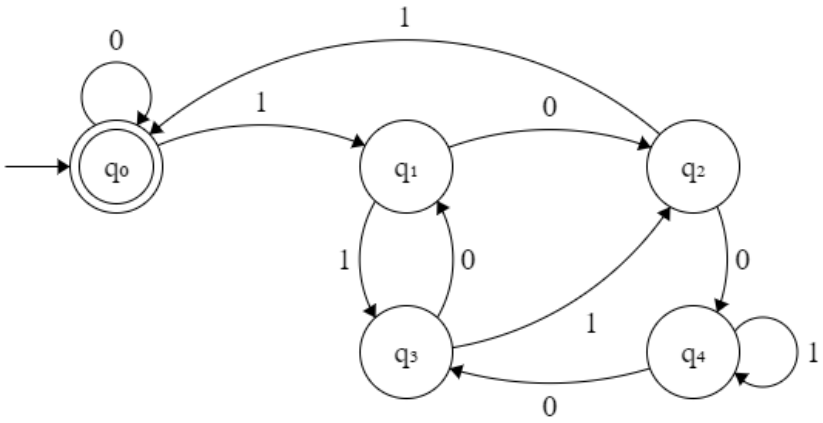
N)



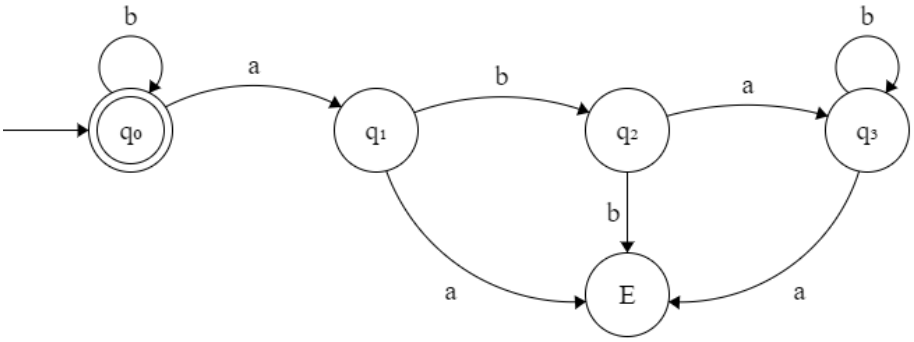


6)

Acepta 000000.... y no deberia

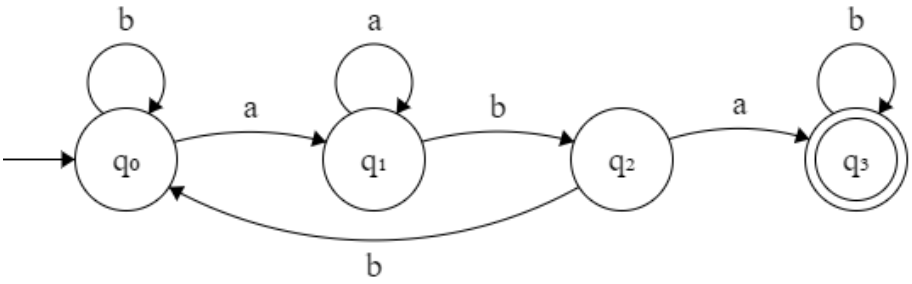


7) Si el lenguaje está formado por las cadenas que contienen una única vez la subcadena **aba** y cualquier cantidad de veces el símbolo **b;** el autómata finito reconocedor es el siguiente:



No cumple con AFD, faltan transiciones en E

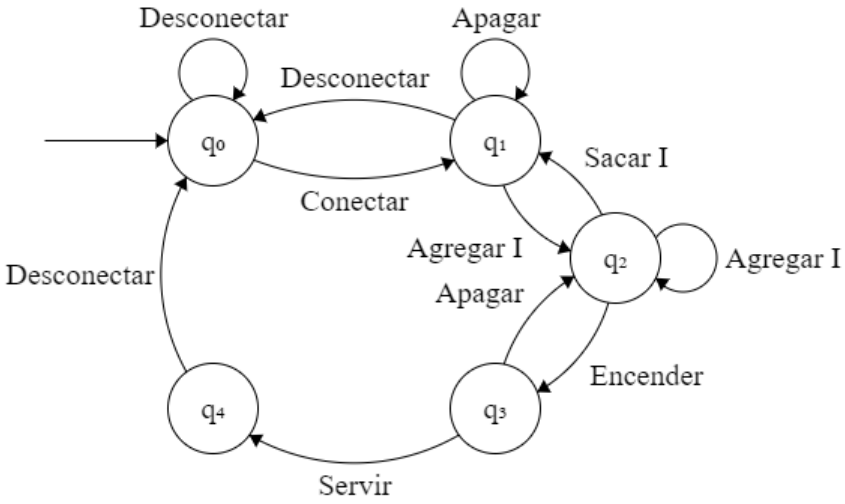
En cambio, si el lenguaje está formado solamente por las cadenas que contienen la subcadena **aba** al menos una vez y cualquier cantidad y combinación de símbolos, el autómata finito reconocedor es el siguiente:



8) EJEMPLO 1:

$A := (Q; \sum; q0; F; \delta)$ $Q = \{q0, q1, q2, q3, q4\}$ $q0 = q0$ $F = \{q0, q1, q2, q3, q4\}$

$\sum = \{\text{Conectar, Desconectar, Encender, Apagar, Sacar I, Agregar I, Servir}\}$



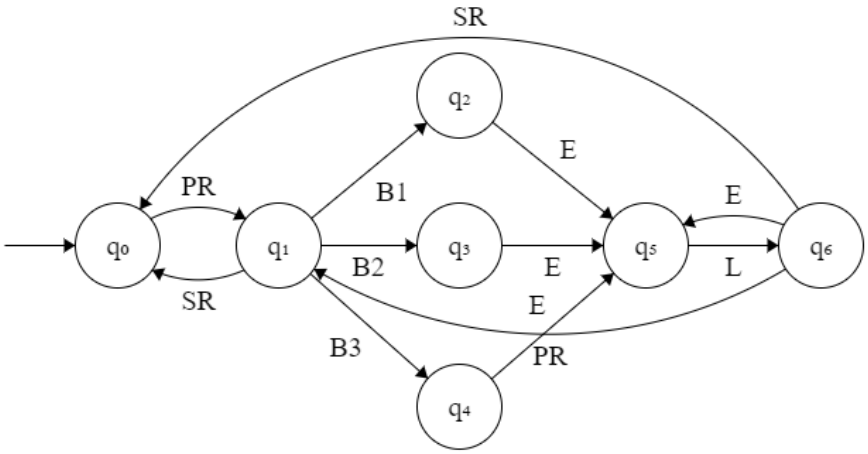
El autómata anterior representa los ***estados de una licuadora*** donde el estado inicial **q_0** *representa la licuadora vacía desenchufada*; el estado **q_1** *la licuadora vacía* pero esta vez *enchufada* a la corriente eléctrica; el estado **q_2** *representa la licuadora enchufada* una vez que se le agregaron los ingredientes, **q_3** la *licuadora encendida* y **q_4** la *licuadora cuando se está sirviendo el licuado*.

EJEMPLO 2:

$A := (Q; \sum; q0; F; \delta)$ $Q = \{q0, q1, q2, q3, q4, q5, q6\}$ $q0 = q0$ $F = \{q0, q1, q2, q3, q4, q5, q6\}$

∑= {PR, SR, B1, B2, B3, E, L}

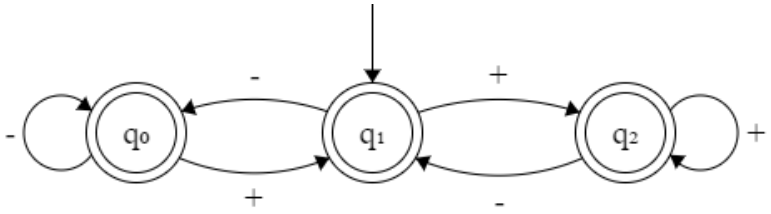
- PR: Poner Ropa.
- SR: Sacar Ropa.
- B1: Seleccionar Botón 1.
- B2: Seleccionar Botón 2.
- B3: Seleccionar Botón 3.
- E: Encender lavarropas.
- L: Lavado finalizado.



El autómata anterior representa los ***estados de un lavarropas,*** donde **q_0** es el lavarropas vacío, **q_1** el lavarropas con ropa sucia, **q_2** – **q_3** – **q_4** los distintos tipos de lavado, **q_5** el estado del lavarropas cuando se seleccionó un tipo de lavado y se está lavando la ropa, **q_6** el estado del lavarropas cuando la ropa ya ha sido lavada.

EJEMPLO 3:

$A := (Q; \sum; q0; F; \delta)$ $Q = \{q0, q1, q2\}$ $\sum = \{-, +\}$ $q0 = q1$ $F = \{q0, q1, q2\}$



El autómata anterior representa los ***cambios de estados que experimenta el agua*** a diferentes temperaturas, donde **q_1** es el *estado inicial* que en la vida real *representa el agua en estado líquido* debido a que éste es el estado en el que se encuentra el agua a temperatura ambiente, **q_0** que *representa el agua en estado sólido* (Hielo) y **q_2** que *representa el agua en estado gaseoso* (Vapor de agua). Los símbolos del alfabeto de entrada – y + *representan una disminución o aumento de la temperatura* respectivamente.

9.

```
from automata.fa.dfa import DFA
dfa = DFA(
    states={'X1', 'X2', 'X3', 'X4'},
    input_symbols={'0', '1'},
    transitions={
        'X1': {'0': 'X1', '1': 'X2'},
        'X2': {'0': 'X2', '1': 'X3'},
        'X3': {'0': 'X3', '1': 'X4'},
        'X4': {'0': 'X1', '1': 'X4'},
    },
    initial_state='X1',
    final_states={'X3'}
)
```

```
prueba= ["1", "0", "111", "000", "010", "011", "10100", "100101", "110110", "111111"]
cadenaaceptada = 0
cadenarechazada = 0
for x in range(0, len(prueba)):
    if dfa.accepts_input(prueba[x]):
        cadenaaceptada += 1
        print("La cadena", prueba[x]+ " fue aceptada porqué el automata se detiene en el estado ", dfa.read_input(prueba[x]))
    else:
        cadenarechazada += 1
        print("La cadena", prueba[x]+ " fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada")
```

```
La cadena 1 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 0 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 111 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 000 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 010 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 011 fue aceptada porqué el automata se detiene en el estado  X3
La cadena 10100 fue aceptada porqué el automata se detiene en el estado  X3
La cadena 100101 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 110110 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada
La cadena 111111 fue rechazada porqué el automata se detiene en un estado de no aceptacion, por lo tanto la cadena es denegada

Process finished with exit code 0
```

```python
minimal_dfa = dfa.minify()
if minimal_dfa == dfa:
    print("El automata utilizado es el mínimo posible")
else:
    print("El automata utilizado no es el mínimo posible")
```

```
El automata utilizado es el mínimo posible

Process finished with exit code 0
```