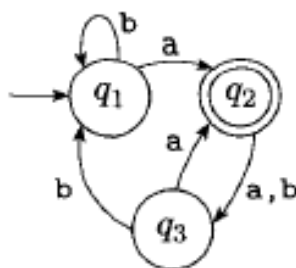


Trabajo Práctico Nº 3

- AFD y AFN.
- Lenguajes regulares.

NOTA: Los ejercicios marcados con * son de resolución y entrega optativa. Los ejercicios marcados con Δ tienen un nivel de dificultad similar a la del parcial.

Ejercicios M_1

1. Para el autómata M_1 determine.

- ¿Cuál es el estado inicial?
- ¿Cuál es el estado de aceptación?
- ¿Qué secuencia de estados sigue el autómata ante la entrada aabb?
- Especifique la descripción formal de M_1

2. Dada la siguiente descripción formal de un autómata, diseñe el diagrama de estados del mismo.

$(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$,

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

3. **Parte 1:** Obtener los diagramas de estado de los autómatas que reconocen los siguientes lenguajes.

- $L = \{w \mid w \text{ comienza con } 1 \text{ y termina con } 0\} \Sigma = \{0,1\}$
- $L = \{w \mid w \text{ contiene al menos tres } 1\} \Sigma = \{0,1\}$
- $L = \{w \mid w \text{ contiene el substring } 0101\} \Sigma = \{0,1\}$
- * $L = \{w \mid w \text{ tiene una longitud de al menos } 3, \text{ y su tercer símbolo es un } 0\} \Sigma = \{0,1\}$

e. $* L = \{w \mid w \text{ tiene longitud impar y comienza con 0, o comienza con 1 y tiene longitud par}\} \Sigma = \{0,1\}$

f. $\Delta L = \{w \mid w \text{ tiene una cantidad par de 0 y 1}\} \Sigma = \{0,1\}$

g. $L = \{w \mid w \text{ no contiene tres b consecutivas}\} \Sigma = \{a,b\}$

h. $\Delta L = \{w \mid w \text{ no contiene las subcadenas aa o bb}\} \Sigma = \{a,b\}$

i. $* L = \{w \mid w \text{ contiene una cantidad impar de 0 y 1}\} \Sigma = \{0,1\}$

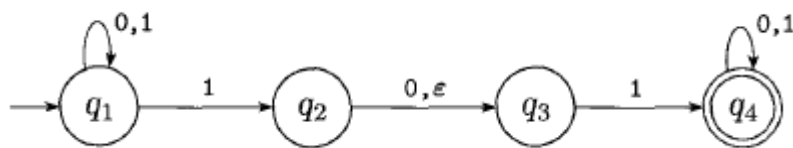
j. $\Delta L = \{w \mid w \in \Sigma^*, y, \text{ si } |w|=5, \text{ entonces contiene al menos dos a's}\} \Sigma = \{a,b\}$

Parte 2: a) Empleando el lenguaje de programación Python y la librería automata-lib valide computacionalmente el diseño de los autómatas de la Parte 1.

b) Verifique computacionalmente el comportamiento de los modelos diseñados en la Parte 1 probando 10 trazas aceptadas y 10 trazas rechazadas.

c) Verifique computacionalmente que los autómatas implementados en a) son los autómatas mínimos.

4. Para el siguiente AFN

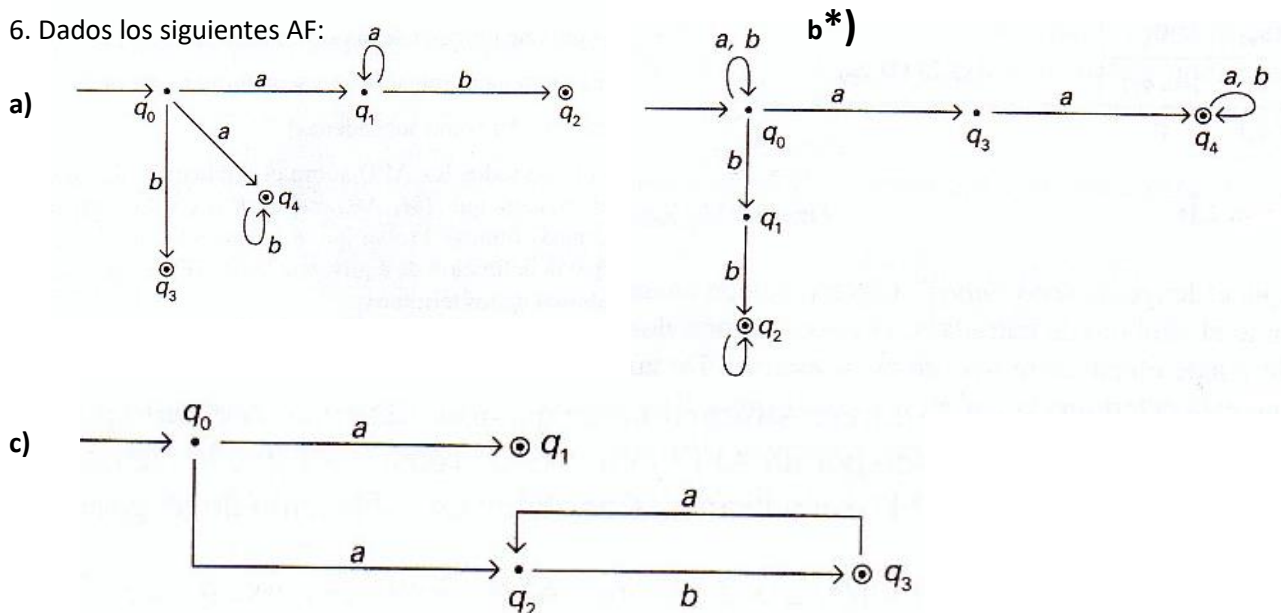


4a. Obtener la descripción formal del mismo.

4b. ¿Qué lenguaje reconoce? Dar ejemplos positivos de cadenas reconocidas y no reconocidas.

5. Diseñe un AFN que reconozca todos los Strings de la forma 0^k donde k es múltiplo de 2 o 3.

6. Dados los siguientes AF:

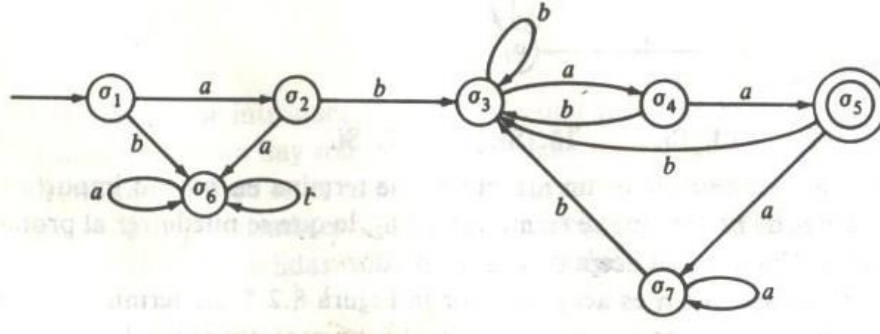


a. Determine si los mismos son determinísticos o no determinísticos.

b. Determine qué lenguaje reconoce cada uno.

7. Δ Para el siguiente autómata:

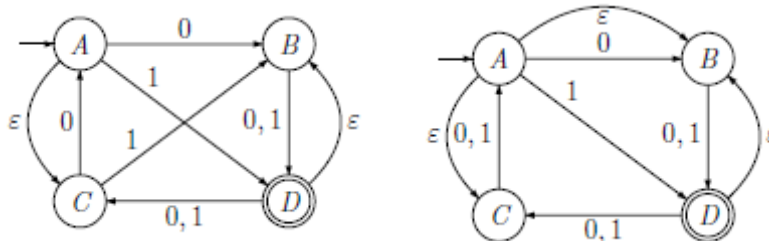
- Especifique: Estado inicial, Estado final y Alfabeto, suponiendo que todos los símbolos están presentes en las transiciones del AF.
- De 3 ejemplos de cadenas reconocidas por el mismo. ¿Qué lenguaje reconoce? (Ej. $L(A_i) = \{w \mid w \dots\}$)



8. Construir autómatas finitos no deterministas que acepten los siguientes lenguajes:

- El lenguaje de las cadenas sobre el alfabeto binario cuyo último símbolo coincida con el primero.
- El lenguaje de las cadenas sobre el alfabeto binario cuyo último símbolo no aparezca con anterioridad.

9. Δ **Parte 1:** Convertir los siguientes autómatas no deterministas en autómatas deterministas



Parte 2: a) Empleando el lenguaje de programación Python y la librería automata-lib valide computacionalmente el diseño de los autómatas de la Parte 1.

b) Verifique computacionalmente que los autómatas implementados en a) son los autómatas mínimos.

c) Verifique computacionalmente la conversión realizada en el punto a)

10. Sean A1 y A2 dos autómatas finitos deterministas, y sean L1 y L2 los lenguajes reconocidos respectivamente. Demostrar las siguientes afirmaciones:

- Existe un autómata A que tiene por lenguaje $L1 \cup L2$.
- Existe un autómata A que tiene por lenguaje $L1 - L2$.
- Existe un autómata A que tiene por lenguaje $L1 \cap L2$.

11. Δ. **Parte 1:** Suponga que los autómatas siguientes representan, cada uno, el comportamiento de dos sistemas distintos; los estados se corresponden con los estados posibles del sistema en cuestión, y las transiciones con las acciones permitidas en cada estado

mencionado. Obtenga un AFD que permita determinar si existen y cuáles son las secuencias de acciones que son válidas en ambos sistemas al mismo tiempo.

Explique la metodología empleada para obtener el autómata.



Parte 2: a) Empleando el lenguaje de programación Python y la librería `automata-lib` valide computacionalmente el diseño del autómata de la Parte 1.

b) Verifique computacionalmente que el autómata implementado en a) es el autómata mínimo.