



Instituto Tecnológico Nacional De México

Instituto Tecnológico De Pachuca

Ingeniería en sistemas computacionales

Semestre: 6 Grupo: A

Materia:

Lenguajes y Autómatas 1

Trabajo:

3.1 Ejercicios AFN

Docente: Ing. Rodolfo Baume Lazcano

Alumna: Melanie Domínguez Figueroa

No. Control: 22200181

Fecha: 24 de Abril de 2024

Los autómatas finitos no deterministas (AFN) son una herramienta fundamental en la teoría de la computación y el análisis de lenguajes formales. A continuación, se presenta una conclusión sobre los AFN:

Definición : Los AFN son una clase de autómatas finitos que no requieren que haya una transición determinista para cada símbolo de entrada en cada estado. Esto significa que un AFN puede tener múltiples trans

Expresividad : Los AFN son equivalentes en expresividad a los autómatas finitos deterministas (AFD), lo que significa que pueden reconocer exactamente los mism

Operaciones : Los AFN pueden ser combinados y manipulados utilizando operaciones como unión, concatenación y estrella de Kleene, al igual que los AFD. Además, los AFN pueden ser minimizados utilizando algoritmos similares a los utilizados para minimizar AFD.

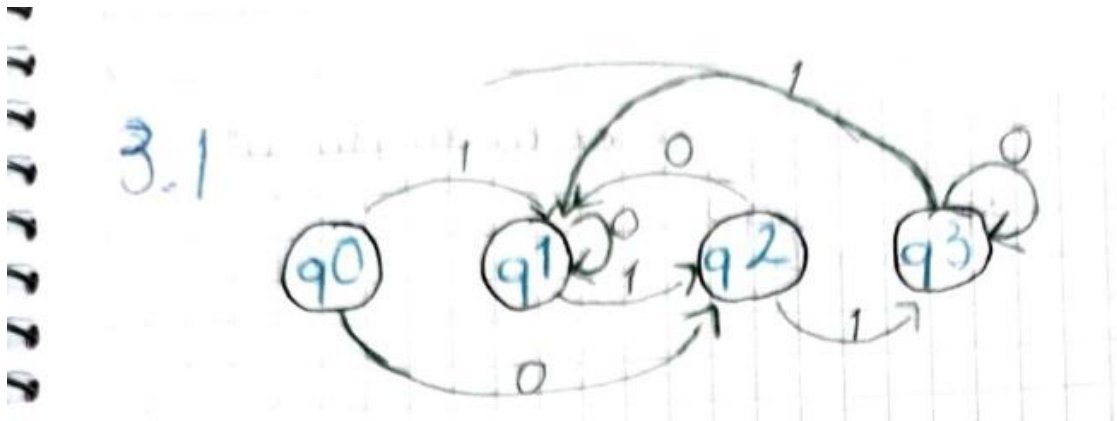
Aplicaciones : Los AFN tienen aplicaciones en diversos campos.

En resumen, los AFN son una herramienta útil y potente en la teoría de la computación y el análisis de lenguajes formales. Aunque tienen limitaciones, son equivalentes en expresividad a los AFD y pueden ser más compactos y fáciles de construir en algunos casos. Los AFN tienen aplicaciones en diversos campos y pueden ser combinados y manipulados utilizando operaciones similares a las utilizadas para los AFD.

Ejercicios Capítulo 3

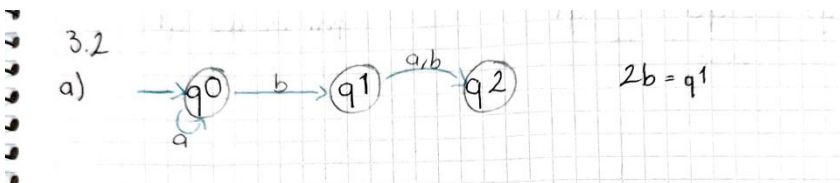
3.1 Construya el diagrama de transición del AFD a partir de la tabla 3.8:

δ	0	1
$\rightarrow^* q_0$	q_2	q_1
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_3	q_1



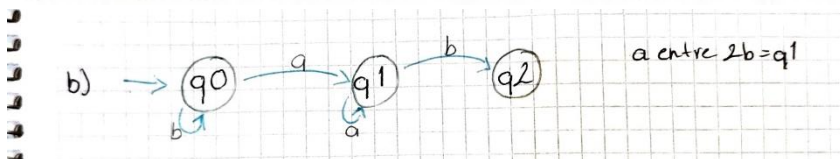
3.2 Para los siguientes ejercicios, construya el diagrama de transición del AFD que acepta a cada uno de los lenguajes sobre el alfabeto $\Sigma = \{ a, b \}$:

a) El lenguaje donde toda cadena tiene exactamente dos bs.



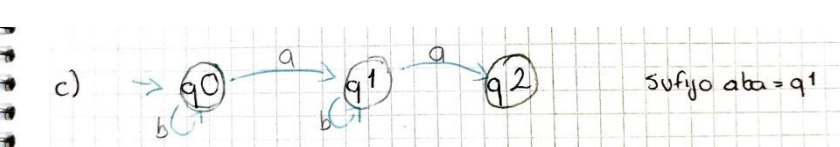
	A	B
q0	q0	q1
q1	q2	q2
q2	q2	q2

b) El lenguaje de las cadenas no vacías, donde toda a está entre dos bs.



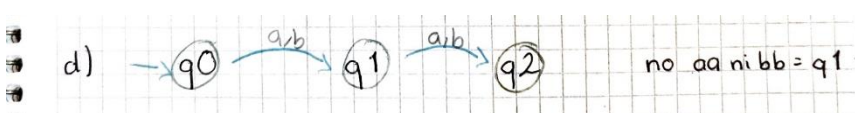
	A	B
q0	q1	q0
q1	q1	q2
q2	q2	q2

c) El lenguaje donde toda cadena contiene el sufijo aba.



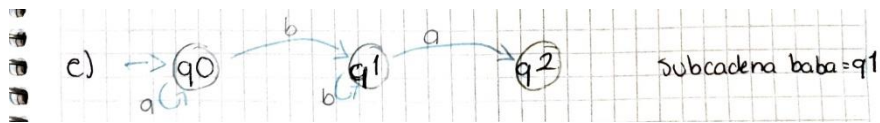
	A	B
q0	q1	q0
q1	q2	q1
q2	q2	q2

d) El lenguaje donde ninguna cadena contiene las subcadenas aa ni bb.



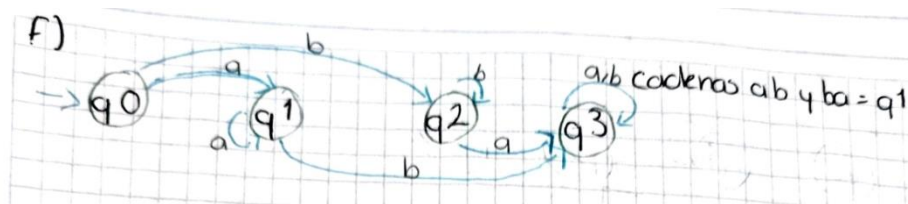
	A	B
q0	q1	q1
q1	q2	q2
q2	q2	q2

e) El lenguaje donde toda cadena contiene la subcadena **baba**.



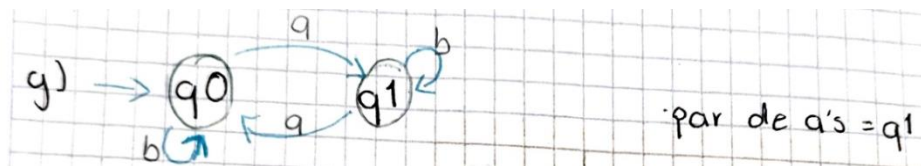
	A	B
q0	q0	q1
q1	q2	q1
q2	q2	q2

f) El lenguaje donde toda cadena contiene por separado a las cadenas **ab** y **ba**.



	A	B
q0	q1	q2
q1	q1	q3
q2	q3	q2
q3	q3	q3

g) Toda cadena es de longitud impar y contiene una cantidad par de as.



	A	B
q0	q1	q0
q1	q0	q1

CONCLUSION:

Los AFN son una herramienta útil y potente en la teoría de la computación y el análisis de lenguajes formales. Aunque tienen limitaciones, son equivalentes en expresividad a los AFD y pueden ser más compactos y fáciles de construir en algunos casos. Los AFN tienen aplicaciones en diversos campos y pueden ser combinados y manipulados utilizando operaciones similares a las utilizadas para los AFD.