



Instituto Politecnico Nacional



ESCOM “ESCUELA SUPERIOR DE CÓMPUTO”

TEORÍA COMPUTACIONAL

PRÁCTICA 3: CONVERSIÓN DE AFN A AFD

PROFA: Luz María Sánchez García

ALUMMNO: Rojas Alvarado Luis Enrique

GRUPO: 2CM11

INTRODUCCION

El propósito de ésta práctica consistió en que a partir de un AFN se convierta en un AFD por medio de la aplicación de JFLAP que posteriormente se le ingresarán cadenas para que sean aceptadas o no en el autómata finito determinista siendo estas cadenas válidas y no válidas respectivamente.

PLANTEAMIENTO DEL PROBLEMA

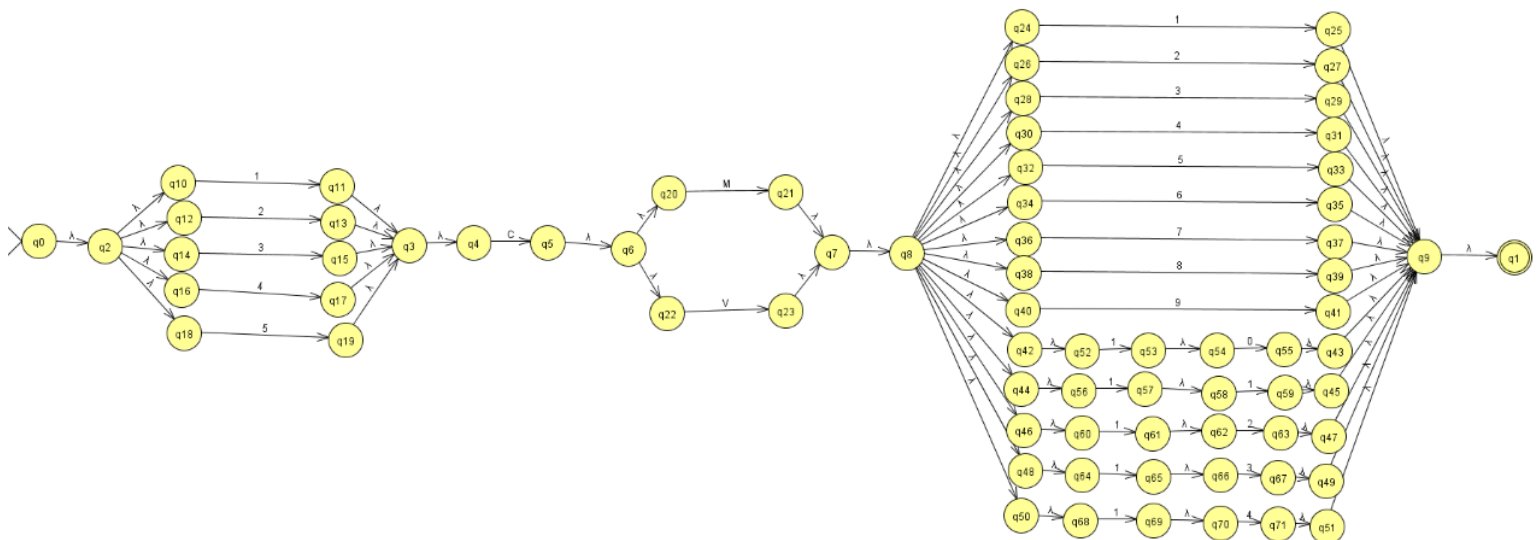
El problema que se presenta es que por medio de una expresión regular y por medio de un autómata finito no determinista realizados en prácticas anteriores se desea convertir el AFN para que sea un AFD por medio del método de subconjuntos que se vió en clase para la conversión de autómatas.

Para darle solución a este problema se optó por utilizar la aplicación de JFLAP para convertir el AFN realizado con la práctica anterior para que sea un AFD a partir de la expresión regular que ya se había implementado.

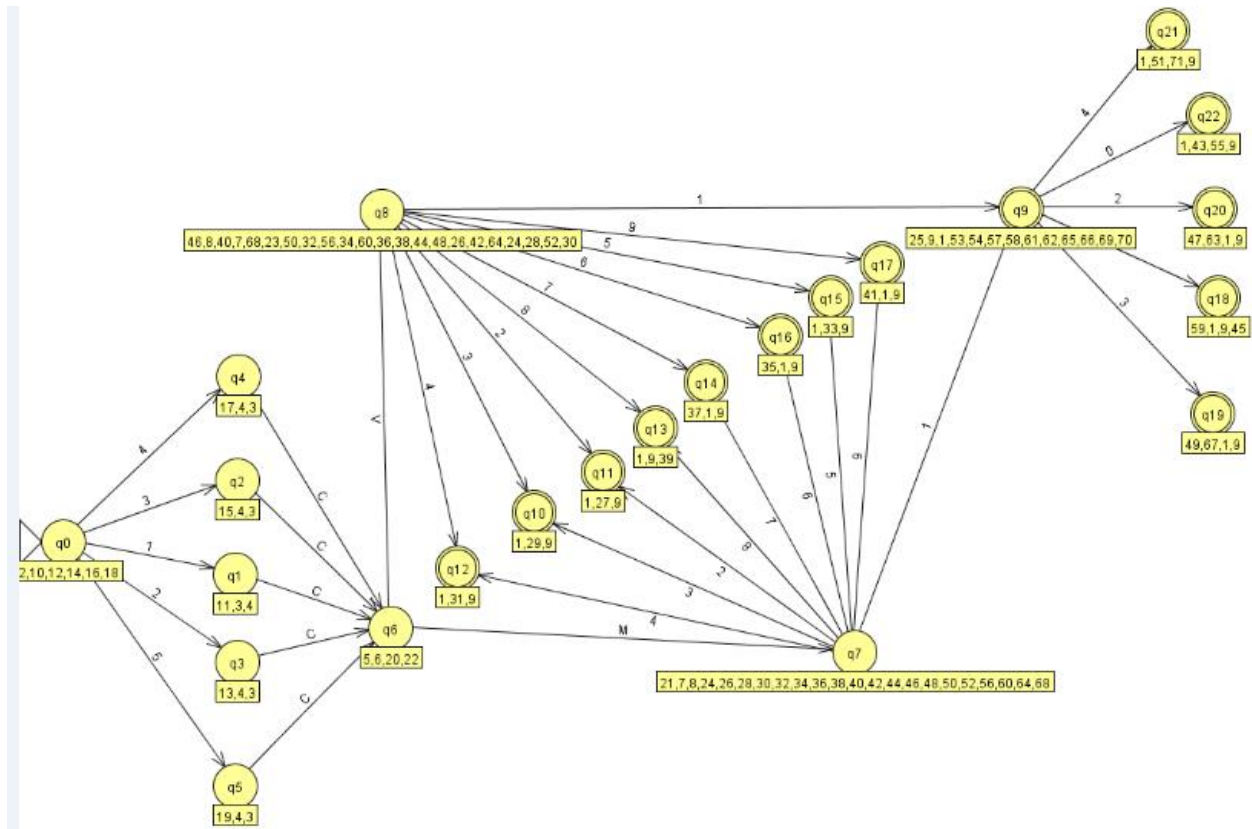
DISEÑO DE LA SOLUCIÓN

Para la solución de este problema dado se eligió como programa para desarrollar éste autómata JFAP. En base a una expresión regular anteriorente desarrollada, por metodo de thompson se realizará el AFN. Una vez realizado simplemente se convierte el autómata a un AFD.

El AFN a convertir es la estructura de un Grupo de ESCOM el cual es el siguiente:



IMPLEMENTACIÓN DE LA SOLUCIÓN



FUNCIONAMIENTO

Se corre el programa en JFALP con la ayuda de JAVA 8:

	2CM2	Accept
	3CM1	Accept
	6CV12	Reject
	2CM11	Accept
	4CV14	Accept
	2CV16	Reject
	2C2F3	Reject
	2KC94	Reject
	6CM99	Reject
	2cm2	Reject
	1CV4	Accept

CONCLUSIONES

En ésta práctica se utilizó la aplicación de JFLAP para la imlementación de la conversión de un AFN a un AFD por medio del método de conjuntos y con la ayuda de JFLAP pudimos hacer la conversión del autómata más sencillamente, aunque en mi opinión la expresión regular que se implementa y después se convierte genera muchos estados y son aun mayores que los estados originales cuando la epresion abarca muchas validaciones y habría que imlementar otro método para la reducción de estados o probar haciendo otro autómata donde los estados sean más fáciles de ineterptretar tanto como para redcir y ser implementados correctamente.

BIBLIOGRAFÍA

- 1) Platzi, 2015, GUuía de expresiones regulares en Python, <https://platzi.com/blog/expresiones-regulares-python/> recuperado el 05/09/2018.
- 2) <http://jflap.org/tutorial/>
- 3) https://www.ecured.cu/Aut%C3%B3mata_finito_determinista