

Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías

Ingeniería en Computación
Teoría de la Computación
Sección D-07

Lepiz Hernández Manuel Alejandro
Código: 214797998



1. Planteamiento del problema

Construir una librería Python que permita crear un Autómata Finito Determinista, y este que solo acepte las cadenas de números binarios que terminen en 00 o 11

2. Objetivo

Aprender y llevar a la practica el uso de un Autómata Finito Determinista, junto con su implementación en JFLAP y Python

3. Justificación

El uso de JFLAP como una herramienta de ayuda, además de la implementación y creación de una librería propia del lenguaje Python, basándose en una librería actual, puede ayudar a otros estudiantes futuros a que puedan construir sus propios autómatas y puedan verificar sus ejercicios.

La herramienta codificada, de cierta manera, servirá como JFLAP, con la ventaja de imprimir en pantalla la tabla de estados de transición generados por el Autómata

4. Marco Teórico

Usando el siguiente lenguaje se construirá el autómata

Table Text Size		
LHS		RHS
S	→	0C
S	→	1A
D	→	0D
B	→	1B
C	→	0D
A	→	1B
D	→	λ
C	→	1A
A	→	0C
D	→	1A
B	→	0C
B	→	λ

El problema en cuestión es más enfocado al desarrollo de la librería, y para su demostración se hará la implementación de un autómata que solo acepte cadenas con 00 o 11 al final.

5. Metodología

Se investigó acerca de como implementar un autómata en Python, a su vez se encontró la problemática de que estos no permitían mostrar el autómata.

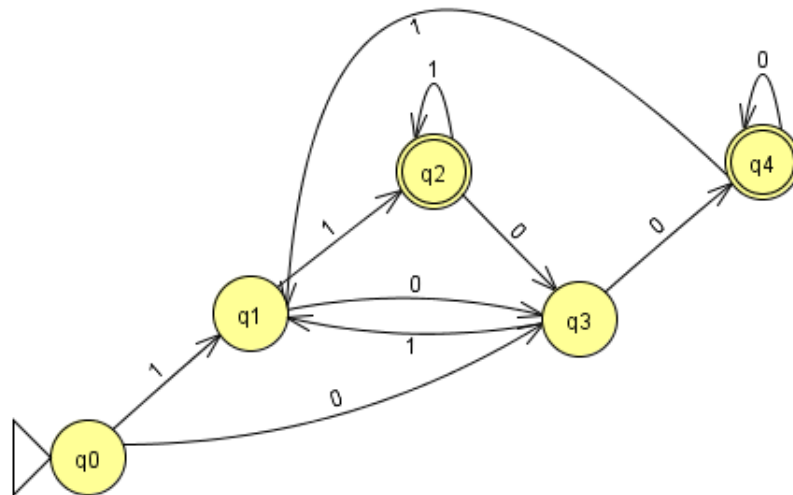
Las implementaciones previas solo contenían casos de prueba.

Se hizo una búsqueda más profunda de como imprimir gráficos desde Python, así el como estaban programados otros autómatas en distintos lenguajes.

Igualmente, no se encontraron implementaciones previas, inclusive en búsquedas en el lenguaje inglés.

Al ver esta problemática se optó por implementar esta librería propiamente.

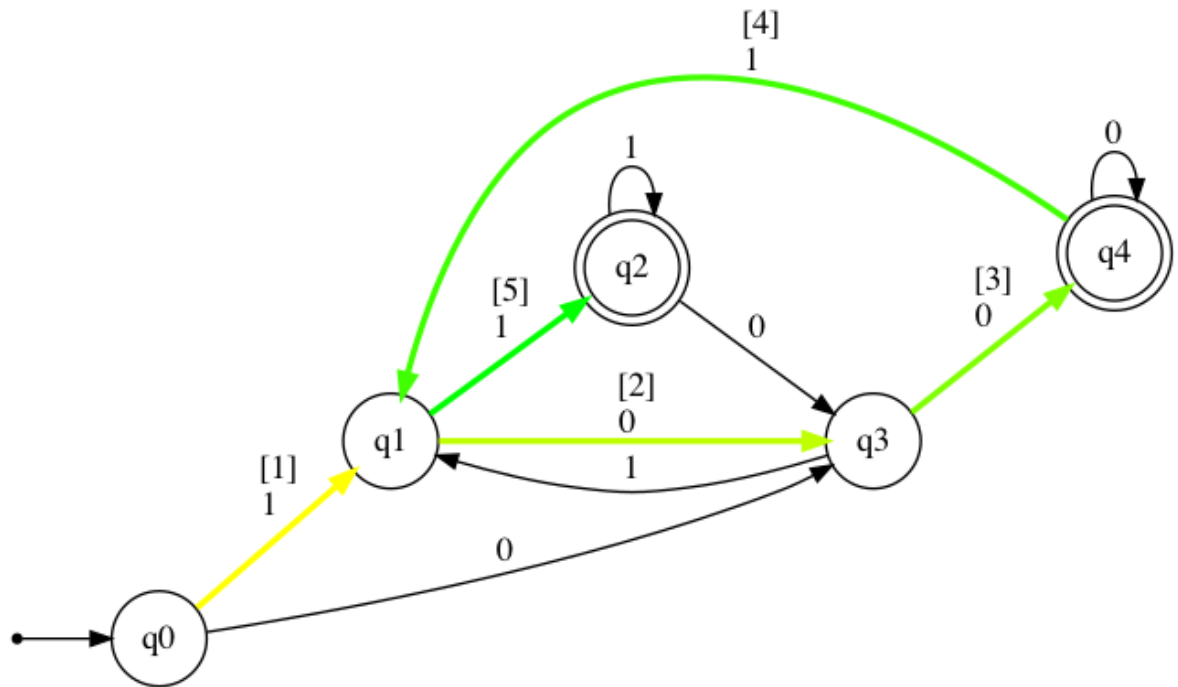
Implementación del autómata en JFlap



6. Pruebas y resultados

Cadena de prueba: "10011"

Teoría de lo que debería de suceder generado por nuestro programa

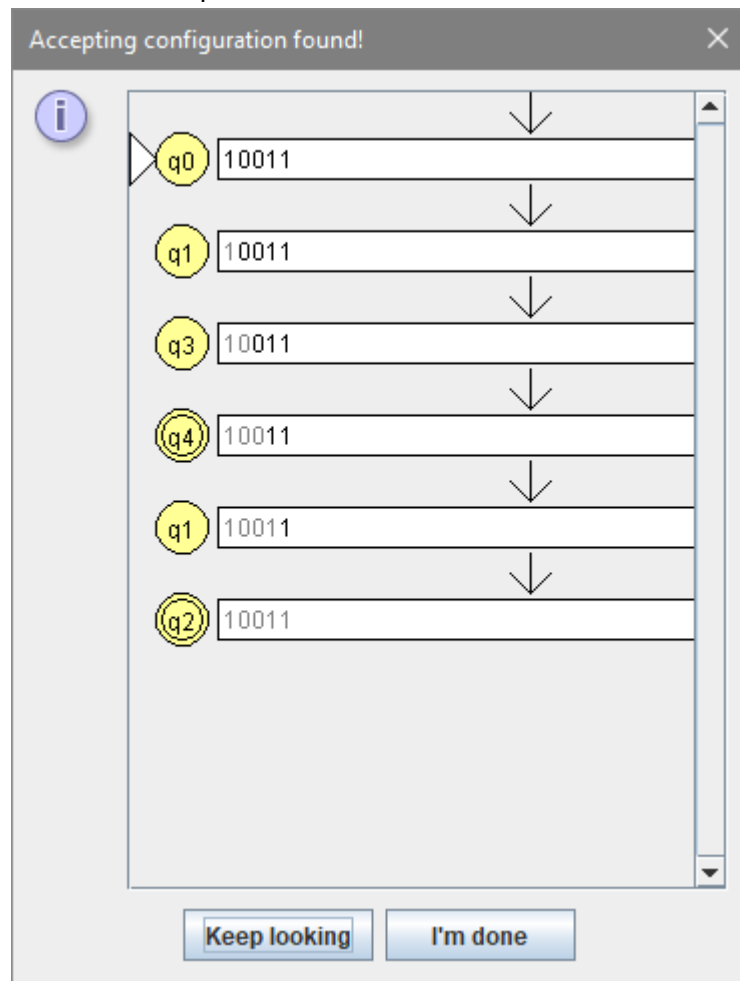


Salida de nuestro programa

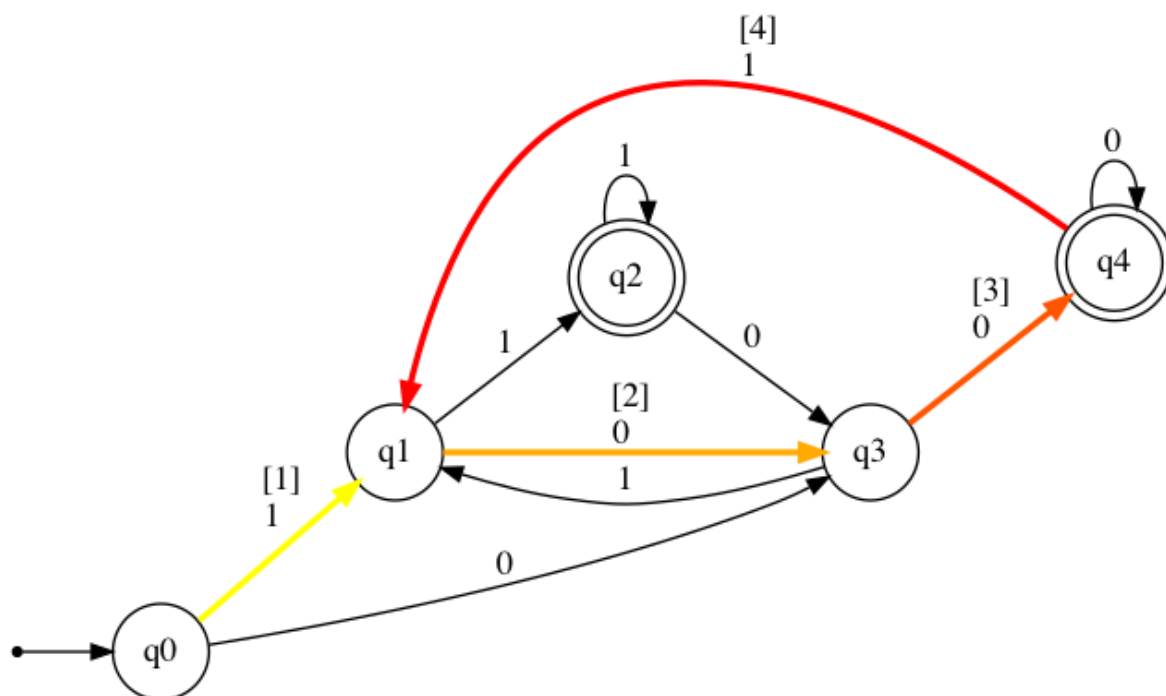
```

[Accepted]
Step: Current state: Input symbol: New state:
1      →q0          1      q1
2      q1          0      q3
3      q3          0      *q4
4      *q4         1      q1
5      q1          1      *q2
  
```

Prueba en JFlap



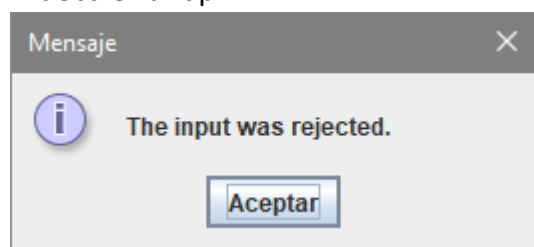
Cadena de prueba "1001"



Salida de nuestro programa

```
[Rejected]
Step: Current state: Input symbol: New state:
1      →q0          1          q1
2      q1           0          q3
3      q3           0          *q4
4      *q4          1          q1
```

Prueba en JFlap



7. Conclusiones

El autómata resultó óptimo, ya que se realizó una buena implementación de la librería.

Puede que con una revisión más detallada pueda realizarse una implementación más óptima

Se cumplió con las expectativas de la implementación de una librería para uso de autómatas y sus estados.