UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA

CURSO

LENGUAJES Y COMPILADORES

Ejercicio AFD

GRUPO "LyCenciados Corruptos"

Bayes Enriquez Eva María Florisa [22200006]

Gonzales Mendieta, Claudio [22200020]

Melendez Blas, Jhair Roussel [22200199]

Pacotaype Chuchon, Diego Alonzo [22200214]

Torres Tineo Cristhian Anthony [22200050]

DOCENTE

Prof. Gelber Christian Uscuchagua Flores

LIMA – PERÚ

2025

Tabla de Contenido

Teoría

Autómatas Determinísticos

Es una máquina que reconoce un lenguaje regular, lo que se conoce formalmente como una 5-tupla (Q, Vt, q0, δ, qF)

Q: Conjunto finito de estados
Vt: Alfabeto de input
q0: Estado inicial
δ: Función de transición
qF: Estado final

Ejercicio 2

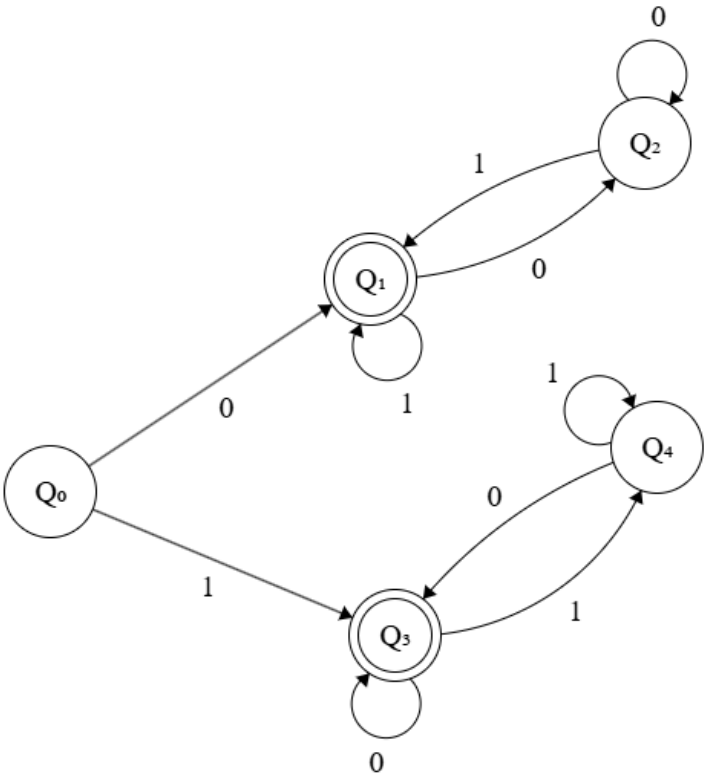
L = { número binario que empieza y termina en dígitos diferentes}

$$\{ 0\ (0 \mid 1)^*\ 1 \mid 1\ (0 \mid 1)^*0 \}$$

Input	Output
01	Cumple
111100	No cumple
111	No cumple
110	Cumple

Diagrama

Se usó el software Finite State Machine Designer, desarrollado por Evan Wallace, para realizar el diagrama de autómatas determinísticos.



Matriz de Transición

Caso: 110

	0	1	$
Q_0	-	Q_3	-
Q_1	-	-	-
Q_2	-	-	-
Q_3	-	Q_4	-
Q_4	Q_3	-	-
Q_3	-	-	Qf

ε = cadena vacía

Tecnologías Usadas

Tecnologías de Desarrollo

- Modelado de Autómatas
- Python

Tecnologías de Gestión

- GitHub
- Excalidraw
- Discord

Código

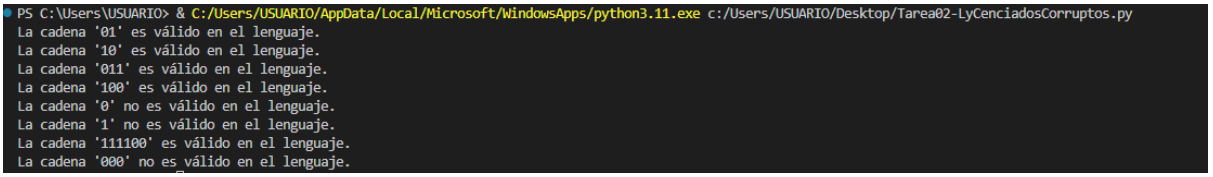
```
import re

# Definir el patrón usando una expresión regular
# 0(0|1)*1 => comienza con 0 y termina con 1
# 1(0|1)*0 => comienza con 1 y termina con 0
pattern = re.compile(r"^(0(0|1)*1|1(0|1)*0)$")

# Función para verificar si una cadena pertenece al lenguaje
def belongLanguage(cadena):
    if pattern.fullmatch(cadena):
        return True
    else:
        return False

# Pruebas
cadenas = ["01", "10", "011", "100", "0", "1", "111100", "000"]
for cadena in cadenas:
    resultado = "es válido" if belongLanguage(cadena) else "no es válido"
    print(f"La cadena '{cadena}' {resultado} en el lenguaje.")
```

Ejecución



Dificultades

La dificultad se centró en el entendimiento completo del funcionamiento de los autómatas determinísticos.

Al comienzo había confusión con respecto a las reglas que debían cumplir los autómatas, y el confundir entre determinísticos y no determinísticos.

Cómo solucionamos las dificultades

Mediante revisión del material brindado en clase, así como la búsqueda de libros y vídeos relacionados al tema dilucidaron nuestras dudas. Importante mencionar el trabajo colaborativo que enriqueció la rápida solución de esta dificultad.

Conclusiones

Este ejercicio nos permitió conectar conceptos teóricos, como las expresiones regulares y los lenguajes formales, con su implementación práctica en Python. Al estructurar el código, comprendimos la importancia de diseñar soluciones claras, reutilizables y eficientes, al tiempo que probamos cómo validar cadenas de forma sistemática. Esto reafirma cómo la programación puede ser una herramienta poderosa para explorar y aplicar principios matemáticos y computacionales.

Referencias

1. Codemath. (2024, February 4). Qué es un Autómata Finito Determinista (AFD) [Video]. YouTube. https://www.youtube.com/watch?v=d9aEE-uLmNE
2. Introducción a la teoría de autómatas, lenguajes y computación (Hopcroft, John E; Ullman, Jeffrey D.).
3. Juancar Molinero. (2014, February 16). Definición Formal de un Autómata Finito Determinista (AFD) [Video]. YouTube. https://www.youtube.com/watch?v=P0AxQvJcN2Q
4. Uscuchagua G. (2025). Notas Magistrales.

Anexos

1. Repositorio de los ejercicios resueltos:
2. Web Page de modelado de autómatas: https://madebyevan.com/fsm/