

Universidad Rafael Landívar
Facultad de Ingeniería
Lenguajes Formales y Autómatas
Sección 01
Ing. Julio Requena



Proyecto Final
Chomsky Classifier AI

Jose Javier Gregorio #1028423

1. Introducción

Chomsky Classifier AI es una herramienta interactiva desarrollada en Python con Pygame que permite:

- Clasificar gramáticas según la Jerarquía de Chomsky
- Analizar y clasificar autómatas, Máquinas de Turing y descripciones mixtas
- Generar diagramas con Graphviz
- Exportar análisis a PDF profesional
- Estudiar mediante Flashcards, Juego de memoria y Carrera de preguntas

2. Requisitos del sistema

Software

- Python 3.10 – 3.12
- Pygame
- Graphviz (instalado y en PATH)
- PyPDF2 (si se usa)
- FPDF (generación de PDF)
- pyperclip (para copiar/pegar)
- Instalar dependencias

```
pip install pygame graphviz fpdf pyperclip
```

Si tienes archivo requirements.txt:

```
pip install -r requirements.txt
```

3. Ejecución del programa

En la carpeta del proyecto:

```
python main.py
```

4. Menú principal

El menú principal contiene:

- Clasificar gramáticas
- Flashcards
- Juego de memoria
- Carrera de quiz
- Salir

Puedes navegar con:

- Mouse
- Flechas
- Enter
- ESC para salir

5. Clasificador de gramáticas

Al seleccionar “Clasificar gramáticas” aparece el editor avanzado.

5.1. Editor de texto (GRAMÁTICA / AUTÓMATA / MT)

Funciones del editor:

Acción	Método
Escribir texto	Normal
Nueva línea	ENTER
Tabulación	TAB
Mover cursor	Flechas ↑ ↓ ← →
Inicio/Fin de línea	HOME / END

Seleccionar todo	CTRL + A
Copiar	CTRL + C
Pegar	CTRL + V
Borrar selección	BACKSPACE / DELETE

6. Ingreso correcto de gramáticas

✓ Formato permitido:

$S \rightarrow aSB \mid ab$

$B \rightarrow b$

$A \rightarrow \epsilon$

Reglas:

- Lado izquierdo = una variable mayúscula
- Flecha \rightarrow
- Producciones separadas por \mid
- ϵ permitido

7. Ingreso de Autómatas

✓ Autómata finito:

Estados: $\{q_0, q_1, q_2\}$

Estado inicial: q_0

Estado de aceptación: $\{q_2\}$

Transiciones:

$$\delta(q_0, a) = q_1$$

$$\delta(q_1, b) = q_2$$

8. Ingreso de Máquinas de Turing

✓ Formato correcto:

Máquina de Turing:

Estados: $\{q_0, q_1, q_{\text{accept}}, q_{\text{reject}}\}$

Alfabeto de cinta: $\{0, 1, B\}$

Alfabeto de entrada: $\{0, 1\}$

Estado inicial: q_0

Estado de aceptación: q_{accept}

Estado de rechazo: q_{reject}

Transiciones:

$$\delta(q_0, 0) = (q_1, 1, R)$$

$$\delta(q_1, 1) = (q_0, 0, L)$$

$$\delta(q_0, B) = (q_{\text{accept}}, B, R)$$

9. Botones principales

Botón	Función
Clasificar	Clasifica gramática / autómata / MT
Generar diagrama	Crea diagrama PNG con Graphviz
Exportar PDF	Genera informe PDF profesional
Cargar ejemplo	Carga una gramática de prueba
Comparar gramáticas	Compara Tipo, lenguajes y características

10. Comparar gramáticas

Formato:

$S \rightarrow aSB \mid ab$

$S \rightarrow aB \mid ab$

$B \rightarrow b$

El sistema analiza:

- Tipo de cada gramática
- Similitudes

- Diferencias
- Si generan o no el mismo lenguaje

11. Generar diagrama

El programa genera:

✓ Gramática:

- Diagrama de derivación
- Con nodos intermedios para producciones largas

✓ Autómata o MT:

- Grafo de estados
- Estados de aceptación → doble círculo
- Flecha de inicio

12. Exportar PDF

El PDF incluye:

- Fecha y hora
- Entrada analizada
- Tipo (0,1,2,3)
- Descripción
- Explicación detallada
- Pasos del análisis
- Observaciones

13. Sistemas de estudio

- ◆ Flashcards

- Tarjetas con conceptos clave
- Avanza con clic o tecla
- ◆ Juego de memoria
 - Cartas con conceptos y definiciones
 - Debes emparejarlas

- ◆ Carrera de quiz
 - Preguntas rápidas
 - Tiempo limitado

14. Errores comunes

Problema	Solución
PDF muestra “Error”	Asegúrate de que la sintaxis esté correcta
No genera diagrama	Instala Graphviz y agrégalo al PATH
Símbolos raros	Usa DejaVuSansMono.ttf
No copia/pega	Instala pyperclip

