

Universidad Rafael Landívar
Facultad de Ingeniería
Ingeniería en Informática y Sistemas
Lenguajes formales y autómatas – Sección 1
Ing. Julio David Requena Duarte



MANUAL DE USUARIO FINAL

Adonis Anthony David Hernández Pérez
Carné: 1086223

Guatemala, 14 de noviembre de 2025

1. Descripción general

Chomsky Classifier AI es una herramienta educativa que analiza una gramática o un autómata (AFD/AP/MT) ingresado por texto y determina el tipo de lenguaje dentro de la Jerarquía de Chomsky (Tipos 0–3). Además, explica paso a paso el razonamiento de la clasificación, genera diagramas (Graphviz/ DOT) y permite exportar reportes.

2. Teoría: Jerarquía de Chomsky

- Tipo 3 Regular**

Lenguajes de patrón simple que no necesitan memoria. Ejemplos: cadenas que terminan en 01 o repiten ab varias veces. Se modela con autómatas finitos y con expresiones regulares.

- Tipo 2 Libre de Contexto**

Permiten anidar y equilibrar. Ejemplos: mismo número de a que de b, uso correcto de paréntesis. Se modela con autómatas con pila. Son más potentes que los regulares, pero no alcanzan todos los patrones **complejos**.

- Tipo 1 Sensible al Contexto**

El contexto influye y las reglas no reducen longitud. Ejemplo: mismo número de a, b y c en ese orden. Se modela con autómatas linealmente acotados. Abarca dependencias más fuertes que una pila simple.

- Tipo 0 Recursivamente Enumerable**

Todo lo que puede reconocer una máquina de Turing. Máxima capacidad expresiva, pero sin garantía de decisión en todos los casos.

- Relación de contención: $\text{Tipo 3} \subset \text{Tipo 2} \subset \text{Tipo 1} \subset \text{Tipo 0}$.**

3. Formatos de entrada por texto

El sistema auto detecta si el contenido es gramática o autómata según el patrón:

A) Gramática (producciones):

$S \rightarrow aA \mid b$

$A \rightarrow aA \mid \epsilon$

B) AFD/AFN (transiciones estado símbolo \rightarrow estado):

AFD

states: q0,q1

start: q0

accept: q1

$q0 \ a \rightarrow q1$

$q0 \ b \rightarrow q0$

$q1 \ a \rightarrow q1$

$q1 \ b \rightarrow q0$

C) AP/PDA (lectura-entrada, tope de pila):

AP

states: p,q

start: p

accept: q

$z_0: Z$

$p, a, Z \rightarrow p, XZ$

$p, \text{eps}, Z \rightarrow q, \text{eps}$

D) MT (lectura/escritura/dirección):

MT

states: q0,q1,qf

start: q0

accept: qf

blank: _

(q0, a) -> (q1, b, R)

(q1, b) -> (qf, b, N)

4. Instrucciones de uso – Modo consola

Guía de instalación y puesta en marcha (CLI + PDF/Streamlit)

1. Requisitos

- Python 3.9 o superior (recomendado 3.10/3.11)
- pip funcionando
- (Opcional) Graphviz para exportar PNG/SVG desde archivos .dot

2. Obtener el proyecto

- Con Git:
`git clone chomsky_classifier_ai`
`cd chomsky_classifier_ai`
- O copia todos los .py en una carpeta llamada chomsky_classifier_ai

3. Instalar dependencias

- Para PDF y UI web:
`python -m pip install --upgrade pip`
`python -m pip install reportlab`
`python -m pip install streamlit`

4. Instalar Graphviz

- Windows: instalar desde la página oficial y agregar dot.exe al PATH

5. Ejecutar en modo consola (CLI)

- En la carpeta del proyecto:

```
python main.py
```

- Opción 1: pega tu gramática o autómata (AFD/AP/MT) y deja una línea vacía para terminar

- Verás: tipo detectado (0–3), máquina equivalente y explicación paso a paso

- Se genera un archivo .dot (gramatica.dot o automata.dot)

6. Generar reporte (PDF o TXT)

- En el menú del programa, opción 2:

- Con reportlab: crea reporte.pdf
- Sin reportlab: crea reporte.txt

7. Interfaz web

- Ejecutar:

```
python -m streamlit run streamlit_app.py
```

- Pegar la entrada y presionar “Clasificar”

- Descargar el .dot y el Reporte (PDF/TXT) desde los botones

5. Ejemplos

Tipo 3:

$S \rightarrow aA$

$A \rightarrow b \mid \epsilon$

Tipo 2:

$S \rightarrow aSb \mid \epsilon$

Tipo 1:

$S \rightarrow aSBC \mid abc$

$CB \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

Tipo 0:

$S \rightarrow AB$

$A \rightarrow aA \mid a$

$B \rightarrow bB \mid b$

$AB \rightarrow a$