

Universidad Rafael Landivar

Facultad de Ingeniería

Lenguajes Formales y Automatas

Sección: 1

Ing.Julio David Requena Duarte

Proyecto Final

Angel Samuel Pérez Cruz

Carné: 1135323

Guatemala 14 de noviembre del 2025

Codigo Fuente (github)

<https://github.com/angel-perez213/ProyectoFinallenguajes.git>

Manual de usuario

Descripcion teorica de la Jerarquia de Chomsky

La Jerarquia de Chomsky es una forma de ordenar los lenguajes formales segun que tan “poderoso” es el modelo que los reconoce y que tan restringidas son sus producciones. Se divide en cuatro niveles:

Tipo 3 – Lenguajes regulares

- Son los lenguajes mas simples.
- Se pueden reconocer con **automatas finitos** (AFD o AFN) y se pueden describir con **expresiones regulares**.
- Las reglas de una gramatica regular suelen ser de la forma:
 - $A \rightarrow aB$
 - $A \rightarrow a$donde A y B son no terminales y a es un terminal.
- La memoria es finita, el automata no puede “contar” infinito.

Ejemplo de gramatica regular:

$S \rightarrow aA$

$A \rightarrow b \mid aA$

Genera cadenas sobre $\{a, b\}$ que terminan en b.

Chomsky Classifier AI

Clasificar gramatica Clasificar automata Comparar gramatica Generador Tutor

Ingresá la gramática:

```
S -> aA
A -> b | aA
```

Diagram...

Clasificar Ver diagrama Generar reporte PDF

Tipo: Tipo 3 - Lenguaje Regular

Explicación:

Revisando si la gramática puede ser Tipo 3 (Regular).
- La producción ' $S \rightarrow aA$ ' es de la forma $A \rightarrow aB$.
- La producción ' $A \rightarrow b$ ' es de la forma $A \rightarrow a$.
- La producción ' $A \rightarrow aA$ ' es de la forma $A \rightarrow aB$.
Todas las producciones cumplen las formas permitidas para una gramática regular.
La gramática cumple las condiciones de Tipo 3 (Regular).

Tipo 2 – Lenguajes libres de contexto

- Se reconocen con un **automata con pila (AP)**.
- Las reglas de produccion tienen **un solo no terminal** a la izquierda:
- $A \rightarrow \text{beta}$
- Permiten describir estructuras anidadas, por ejemplo parentesis balanceados o el clasico $a^n b^n$.

Ejemplo:

$S \rightarrow aSb$

$S \rightarrow ab$

Genera cadenas de la forma $a^n b^n$, con $n \geq 1$. Esta gramatica ya no es regular, pero si es libre de contexto.

Chomsky Classifier AI

Clasificar gramatica Clasificar automata Comparar gramatica Generador Tutor

Ingresar la gramatica:

$A \rightarrow \text{beta}$

Diagram...

Clasificar Ver diagrama Generar reporte PDF

Tipo: Tipo 2 - Lenguaje Libre de Contexto

Explicacion:

Revisando si la gramatica puede ser Tipo 3 (Regular).
- La produccion 'A -> beta' tiene longitud mayor que 2 en el lado derecho.
Conclusion: la gramatica no es Tipo 3.
Revisando si la gramatica puede ser Tipo 2 (Libre de contexto).
Todas las producciones tienen un solo no terminal en el lado izquierdo.
La gramatica cumple las condiciones de Tipo 2 (Libre de contexto).

Tipo 1 – Lenguajes sensibles al contexto

- Se reconocen con un **automata linealmente acotado**.
- Las producciones cumplen que la parte derecha nunca es mas corta que la izquierda:
- $|\alpha| \leq |\beta|$
- El lado izquierdo puede tener varios simbolos (ahi entra el “contexto”).

Ejemplo de produccion sensible al contexto:

$AB \rightarrow BA$

Aqui no solo importa A y B, sino en qué orden aparecen.

Chomsky Classifier AI

Clasificar gramatica Clasificar automata Comparar gramatica Generador Tutor

Ingresa la gramatica:

$AB \rightarrow BA$

Diagram... — □ ×

Clasificar Ver diagrama Generar reporte PDF

Tipo: Tipo 1 - Lenguaje Sensible al Contexto

Explicacion:

Revisando si la gramatica puede ser Tipo 3 (Regular).
- La produccion 'AB -> BA' no tiene un solo no terminal en el lado izquierdo.
Conclusion: la gramatica no es Tipo 3.
Revisando si la gramatica puede ser Tipo 2 (Libre de contexto).
- La produccion 'AB -> BA' no tiene exactamente un no terminal en el lado izquierdo.
Conclusion: la gramatica no es Tipo 2.
Revisando si la gramatica puede ser Tipo 1 (Sensible al contexto).
Todas las producciones respetan la condicion de longitud para Tipo 1.
La gramatica cumple las condiciones de Tipo 1 (Sensible al contexto).

Tipo 0 – Lenguajes recursivamente enumerables

- Son los lenguajes mas generales de la jerarquia.
- Se reconocen con una **Maquina de Turing**.
- No hay restricciones fuertes sobre las producciones, cualquier $\alpha \rightarrow \beta$ es valido.
- Incluyen todo lo que una computadora “ideal” podria reconocer (aunque a veces no termine).

Instrucciones de uso del sistema

2.1. Como ejecutar el programa

1. Abrir **Visual Studio Code** en la carpeta del proyecto chomsky_classifier_ia.
2. Verificar que VS Code este usando el interprete de **Python 3** (abajo a la derecha).
3. Abrir el archivo main.py.
4. Dar clic en el boton de **Run Python File** (el triangulito ►) o ejecutar con F5.
5. Se abre una ventana de escritorio llamada “**Chomsky Classifier AI**” con varias pestañas.

2.2. Pestaña: “Clasificar gramatica”

Paso a paso:

1. En el cuadro grande de texto, escribir la gramatica. Ejemplo:
 2. $S \rightarrow aA$
 3. $A \rightarrow b \mid aA$
4. Dar clic en “**Clasificar**”.
 - Abajo aparece:
 - El tipo de lenguaje (por ejemplo: “Tipo 3 - Lenguaje Regular”).
 - Una explicacion linea por linea de por que se clasifica asi.
5. Opcional:
 - “**Ver diagrama**”: abre una ventana con un grafo que representa las producciones.
 - “**Generar reporte PDF**”: permite guardar un PDF con:
 - La gramatica.
 - El tipo detectado.
 - La explicacion.
 - El diagrama generado.
 -

2.3. Pestaña: “Clasificar automata”

1. En el cuadro de texto pegar el automata en formato **JSON**. Ejemplo de AFD:

```
1. {
2.   "tipo": "AFD",
3.   "estados": ["q0", "q1"],
4.   "alfabeto": ["a", "b"],
5.   "estado_inicial": "q0",
6.   "estados_finales": ["q1"],
7.   "transiciones": [
8.     { "origen": "q0", "simbolo": "a", "destino": "q1" },
9.     { "origen": "q0", "simbolo": "b", "destino": "q0" },
10.    { "origen": "q1", "simbolo": "a", "destino": "q1" },
11.    { "origen": "q1", "simbolo": "b", "destino": "q0" }
12.  ]
13. }
```
14. }

15. Dar clic en “**Clasificar automata**”.

- El sistema dice si el modelo es AFD, AP, MT, etc. y a que tipo de lenguaje corresponde (3, 2 o 0).
- “**Ver diagrama**”: genera el grafo de estados y transiciones.
- “**Generar reporte PDF**”: igual que en las gramáticas, pero con datos del automata.

The screenshot shows the Chomsky Classifier AI web application. At the top, there's a navigation bar with links: 'Chomsky Classifier AI', 'Clasificar gramática', 'Clasificar automata' (which is currently selected), 'Comparar gramática', 'Generador', and 'Tutor'. Below the navigation is a text input field labeled 'Automata en formato JSON:' containing the JSON code provided in the previous step. At the bottom of the input field is a button labeled 'Clasificar automata'. To the right of the input field, there are three buttons: 'Ver diagrama', 'Generar reporte PDF', and another 'Clasificar automata' button. Below these buttons, the text 'Tipo: Tipo 3 - Lenguaje Regular' is displayed. At the very bottom, there's a note: 'El automata se identifica como AFD o AFN. Los automatas finitos reconocen lenguajes regulares (Tipo 3)'.

2.4. Pestaña: “Comparar gramatica”

Sirve para ver si dos gramáticas parecen generar el mismo lenguaje, hasta cierta longitud de cadenas.

1. Escribir una gramática en el cuadro de la izquierda (**Gramática 1**).
2. Escribir otra gramática en el cuadro de la derecha (**Gramática 2**).
3. Elegir en el spinbox la **longitud maxima** de cadenas que se van a generar para comparar (por ejemplo 4).
4. Dar clic en “**Comparar**”.
5. El sistema genera cadenas hasta esa longitud y dice si:
 - o No encontró diferencias (posible equivalencia), o
 - o Encontró cadenas que una gramática acepta y la otra no.

Es una comparación aproximada (no prueba equivalencia total, solo hasta la longitud indicada).

The screenshot shows the Chomsky Classifier AI interface with the 'Comparar gramatica' tab selected. There are two input fields for grammars:

- Gramatica 1:** Contains the grammar rule $S \rightarrow aSb \mid ab$.
- Gramatica 2:** Contains the grammar rule $S \rightarrow aSb \mid ab$.

A spinbox labeled "Longitud maxima de cadenas a comparar:" shows the value "4". Below the spinbox is a "Comparar" button. At the bottom of the window, a message states: "Posible equivalencia: los lenguajes generados coinciden hasta la longitud maxima indicada."

2.5. Pestaña: “Generador”

Esta pestaña la uso para generar ejemplos de prueba de forma automatica.

- Parte de arriba:
 1. Selecciono el tipo de gramatica (0, 1, 2 o 3) en el combobox.
 2. Presiono “**Generar gramatica**”.
 3. En el cuadro de abajo aparece una gramatica aleatoria de ese tipo.
- Parte de abajo:
 1. Selecciono el tipo de automata de ejemplo (0, 2 o 3).
 2. Presiono “**Mostrar automata de ejemplo**”.
 3. El sistema muestra un JSON de un automata representativo de ese nivel.

Con esto puedo copiar esos ejemplos a las otras pestañas y probar el clasificador.

Chomsky Classifier AI

Clasificar gramatica Clasificar automata Comparar gramatica Generador Tutor

Tipo de gramatica a generar (0,1,2,3):

3

Generar gramatica

S -> bA
A -> b | aA

Tipo de automata de ejemplo (0,2,3):

3

Mostrar automata de ejemplo

```
{  
  "tipo": "AFD",  
  "estados": [  
    "q0",  
    "q1"  
  ],  
  "alfabeto": [  
    "a",  
    "b"  
  ]  
}
```

2.6. Pestaña: “Tutor”

Este es el modo tipo quiz para practicar.

1. Presiono “Nuevo ejercicio”.

- El sistema genera una gramática o automata y muestra el enunciado.

2. Leo el enunciado y, en el combobox, selecciono el tipo que yo creo que es:

- Tipo 3 – Regular
- Tipo 2 – Libre de contexto
- Tipo 1 – Sensible al contexto
- Tipo 0 – Recursivamente enumerable

3. Presiono “Ver resultado”.

4. El sistema indica:

- Si la respuesta es correcta.
- Si es incorrecta, explica cual era el tipo correcto y por que.

De esta forma, el programa no solo clasifica, sino que funciona como herramienta de estudio.

The screenshot shows the Chomsky Classifier AI application window. At the top, there is a navigation bar with tabs: Clasificar gramática, Clasificar automata, Comparar gramática, Generador, and Tutor. The 'Tutor' tab is currently active, indicated by a blue border around its tab. Below the tabs, there is a button labeled 'Nuevo ejercicio'. The main area contains a text input field with the placeholder text: "Clasifica el lenguaje reconocido por el siguiente automata (en JSON):". Below this, there is a JSON object representing a Deterministic Finite Automaton (DFA):

```
{
  "tipo": "AFD",
  "estados": [
    "q0",
    "q1"
  ],
  "alfabeto": [
    "a",
    "b"
  ],
  "estado_inicial": "q0",
  "estados_finales": [
    "q1"
  ],
  "transiciones": [
    {
      "origen": "q0",
      "simbolo": "a",
      "destino": "q1"
    }
  ]
}
```

Below the JSON input field, there is a label "Selecciona tu respuesta:" followed by a dropdown menu containing the option "Tipo 3 - Regular". At the bottom of the window, there is a "Ver resultado" button.

Ejemplos y casos de prueba

3.1. Gramaticas

Ejemplo 1 – Gramatica tipo 3 (Regular)

Entrada en la pestaña “Clasificar gramatica”:

$S \rightarrow aA$

$A \rightarrow b \mid aA$

Resultado esperado:

- Tipo: **Tipo 3 – Lenguaje Regular.**
- Explicacion (resumen):
 - Todas las producciones tienen un solo no terminal a la izquierda.
 - La parte derecha es de la forma aB o a , lo cual cumple las formas tipicas de las gramaticas regulares.

Ejemplo 2 – Gramatica tipo 2 (Libre de contexto)

$S \rightarrow aSb$

$S \rightarrow ab$

Resultado esperado:

- Tipo: **Tipo 2 – Lenguaje Libre de contexto.**
- Explicacion:
 - El lado izquierdo de las producciones es un solo no terminal.
 - La gramatica no es regular porque la produccion $S \rightarrow aSb$ genera dependencia entre la cantidad de a y b .

Ejemplo 3 – Gramatica tipo 1 (Sensible al contexto)

$S \rightarrow aSBC \mid abc$

$CB \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bC \rightarrow bc$

cC -> cc

Resultado esperado:

- Tipo: **Tipo 1 – Sensible al contexto.**
- Explicacion:
 - Hay producciones con mas de un simbolo en la parte izquierda (por ejemplo CB -> BC).
 - Se respeta la condicion $|\alpha| \leq |\beta|$.
 - Por eso no es tipo 2, pero si cumple las reglas de tipo 1.

Ejemplo 4 – Gramatica tipo 0

S -> AB

A -> aA | a

AB -> A

Resultado esperado:

- Tipo: **Tipo 0 – Recursivamente enumerable.**
- Explicacion:
 - Aparecen producciones como AB -> A que pueden reducir la longitud y no respetan las restricciones de tipo 1.
 - Al no cumplir los requisitos de los tipos 1, 2 o 3, se clasifica como tipo 0.