UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

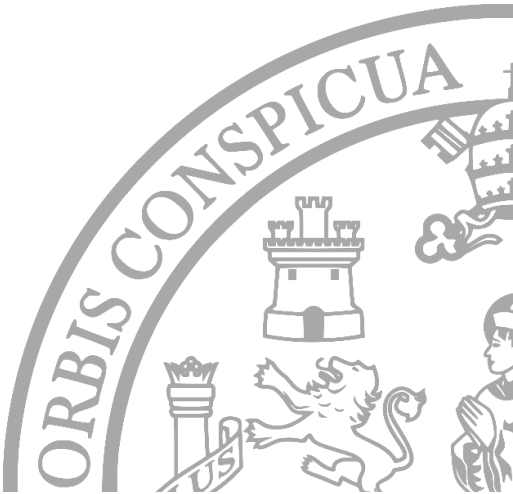
LENGUAJES FORMALES Y PROGRAMACIÓN

PROYECTO 2

JOAQUIN EMMANUEL ALDAIR COROMAC HUEZO

201903873

GUATEMALA, 25 DE ABRIL DE 2021



MANUAL TECNICO

Lógica Programa

➢ **Archivo(main.py)**

- Funciones
  - def main():

Función principal para ejecutar el programa y llama a la función Menu del archivo proyecto.py

➢ **Archivo(proyecto.py)**

- Clases
  - class gramática:

Cada gramática tipo 2 se almacena en un objeto “gramática” con los atributos nombre, terminales, noterminales, inicial y producciones.

- Funciones
  - def menú():

Menu principal del programa.

  - def glc():

Verifica las gramáticas del archivo de entrada, si consiste en una gramática tipo 2 crea el objeto “gramática” y luego lo guarda en una lista.

  - def grafica():

Crea el archivo .dot de graphviz y escribe sobre este para generar la grafica del autómata de pila equivalente para las gramáticas que desea el usuario.

  - def HTML_grafica():

Toma la imagen generada del AP equivalente y la muestra en una página HTML en el navegador.

➢ **Archivo(grafo.dot)**

- Archivo generado para cada AP equivalente de las gramáticas libres de contexto.

Explicación de gramática tipo 2

Una gramática ("G") desde el punto de vista de la teoría de autómatas es un conjunto finito de reglas que describen toda la secuencia de símbolos pertenecientes a un lenguaje específico L. Dos gramáticas que describan el mismo lenguaje se llaman gramáticas equivalentes.

Una gramática es una estructura algebraica formada por cuatro elementos fundamentales:

G = { NT, T, S, P }

Donde:

NT es el conjunto de elementos No Terminales

T es el conjunto de elementos Terminales

S es el Símbolo inicial de la gramática

P es el conjunto de Reglas de Producción

Tipo 2 o "libre de contexto"

$$x \to y$$
$$x \in NT$$
$$y \in (NT/T)^*$$

x puede ser reemplazado por *y* si *x* pertenece a los símbolos **No Terminales** e *y* es un **Terminal** o **No Terminal**, incluyendo la cadena vacía.

Máquinas que los pueden leer:

Máquinas que los aceptan: Autómata a Pila

Para generar los autómatas de pila se utilizó el método 2.2,que dice:

para cada gramática independiente del contexto, existe un autómata de pila M tal que

L(G)=L(M).

Dada una gramática G independiente del contexto es posible construir un autómata de pila M de la manera siguiente:

1. Designe el alfabeto del autómata M como los símbolos terminales de G, y los símbolos de pila de M como los símbolos terminales y no terminales de G, junto con el símbolo especial #.

2. Designe los estados del autómata M como i,p,q,f donde i es el estado inicial y f es el único estado de aceptación.

3. Introduzca la transición (i, λ, λ;p,#)

4. Introduzca la transición (p, λ, λ;q,S) donde S es el símbolo inicial de G.

5. Introduzca una transición de la forma (q,λ,N;q,w) para cada regla de reescritura N→w en G.

6. Introduzca una transición de la forma (q,x,x;q,λ) para cada terminal de x de G (es decir, para cada símbolo del alfabeto de M).

7. Introduzca la transición (q, λ,#;f, λ)

Explicación de AFD

Para reconocer las gramáticas tipo 2 del archivo de entrada estas debían tener las siguientes caracteristicas:

- Nombre en la primera linea
- En la segunda linea deben ir los noterminales, terminales y noterminal Inicial separados por punto y coma
- Las producciones deben ser de la forma que acepten las libres de contexto
- En la ultima linea debe ir un Símbolo asterisco “*”

Para saber si una producción es de la forma de las libres de contexto del lado derecho, se observa que debe haber mas de 2 terminales o no terminales

Especificaciones

Especificaciones de Editor de Texto, versión Python y Graphviz

Visual Studio Code

python.3.8.1

Graphviz 2.46.0

Especificaciones de computadora

Procesador: Intel(R) Core™ i7-9750H CPU@

2.60 GHz 2.59GHz

RAM instalada: 16.0GB (15.9 GB usable)

Tipo de sistema: Sistema operativo de 64 bits,

Procesador basado en x64

Windows 10.

Control de versiones

Para el proyecto se trabajó con GitHub para controlar las versiones durante su realización.

Enlace Repositorio.

https://github.com/jeach27/-LFP-Proyecto2_201903873.git