

Manual Técnico

Byron Estuardo Solís González

201906588

Primer Semestre – Abril 2021

INTRODUCCIÓN

Se desarrolló una aplicación en consola utilizando en lenguaje de programación Python, esta permite ingresar gramáticas libres del contexto con las que se pueden realizar las siguientes opciones: Crear la representación gráfica del autómata de pila que reconocería la gramática ingresada, además permite verificar si una cadena de caracteres ingresada por el usuario pertenece o no al lenguaje que reconoce la gramática seleccionada a través de dos caminos; recorrido gráfico donde se muestran todas las iteraciones y se señala el proceso llevado a cabo en el autómata a través de su diagrama y otro a través de una tabla donde se especifican las iteraciones, transacciones y contenido de la pila y la entrada en el proceso de evaluación de la cadena.

Para la creación de grafos se utilizó Graphviz, una librería de Python basada en Dot especializada en esta tarea; Por otro lado para la valuación de pertenencia de las cadenas de caracteres se utiliza un autómata de pila generado de forma dinámica creado en base a la gramática que el usuario elige.

Con esto se llegó a una aplicación simple con varias formas de representar y entender la valuación de cadenas de caracteres a través de autómatas de pila basados en gramáticas de tipo 2.

DATOS TECNICOS

- Lenguaje utilizado para la aplicación : Python
- IDE utilizado: Visual Studio Code
- Lenguaje utilizado para reportes: HTML

FLUJO DE TRABAJO

1. Ingreso de datos

La aplicación se alimenta de los datos que el usuario le proporciona a través de archivos de entrada con extensión .glc con una estructura definida que contiene la información de las gramáticas libres del contexto

a. Archivos de entrada

Los archivos de entrada pueden contener varias gramáticas, como delimitador de estas se utiliza el símbolo “*”, En la primera línea de la gramática se encuentra el nombre de esta, en la segunda los símbolos No Terminales, Terminales y el Terminal inicial separados por “;,” Y en el resto de líneas se ubican las producciones, para representar que un símbolo tiene más de una producción se utiliza el símbolo “|”.

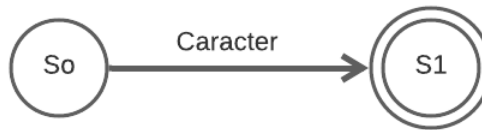
```
Grm1
S,A,B,C;a,b,z;S
S->A
A->a A a
A->B
B->b B b
B->C
C->z C
C->z
*
```

b. Cadenas para valuación

Para la recolección de los símbolos en las cadenas de caracteres que el usuario ingresa para comprobar si pertenecen o no a una gramática se utilizó un sencillo DFA.

i. DFA

El DFA empleado acepta cualquier carácter individual, guardándolo como un token de sí mismo.

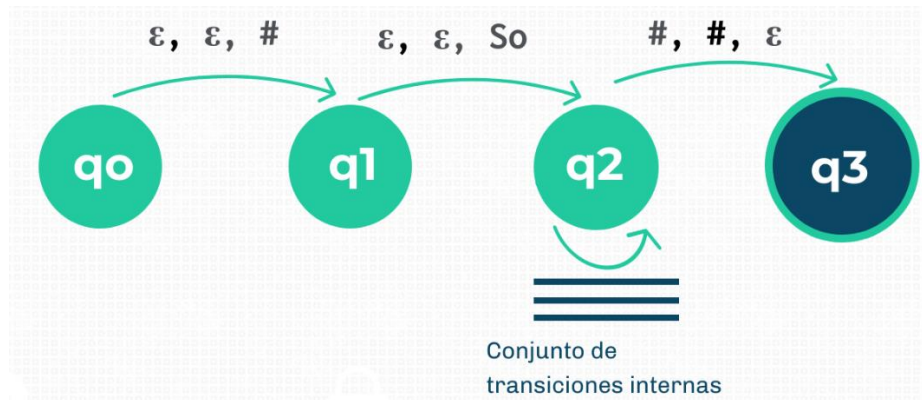


2. Funcionalidades:

- a. Cargar archivo: Esta opción permite cargar un archivo de entrada con extensión .glc que contiene la información de las gramáticas libres del contexto
- b. Mostrar información general de la gramática:
Esta opción del menú muestra todos los nombres de gramáticas que se encuentran actualmente en el sistema para que el usuario pueda elegir una. Cuando el usuario elija una gramática, inmediatamente se muestra la información de la gramática en la consola. Estos datos son:: Nombre de la gramática tipo 2, Simbolos No terminales, Simbolos Terminales, Simbolo No terminal inicial y las producciones de la gramática.
- c. Generar autómatas de pila equivalente: En esta opción el usuario elige una gramática cargada en el sistema y a continuación genera un reporte en html que muestre el autómata de pila equivalente a través de un grafo, este se guarda de forma automática con el nombre de la gramática, anteponiendo AP_.
- d. Reporte de recorrido: El usuario elige una gramática de las cargadas al sistema. Finalizada la elección, se solicita al usuario el ingreso de una cadena de entrada para que sea validada con el autómata. Esta opción permite generar un reporte en html que contiene a detalle cada una de las iteraciones realizadas para validar la cadena.
- e. Reporte en tabla: El usuario elige una gramática de las cargadas al sistema. Finalizada la elección, se solicita al usuario el ingreso de una cadena de entrada para que sea validada con el autómata. Esta opción genera como resultado un reporte en html con una tabla de resumen de las iteraciones realizadas para validar la cadena.

3. Autómata de Pila

Para la creación de los autómatas de pila se optó por utilizar el método con gramática, siendo estas las que ingresa el usuario. Debido a esto se crearon los cuatro estados estándar del autómata, implementando el uso de la pila en cada uno de ellos



1. Transiciones internas

A través de estas se lleva a cabo la validación de la cadena, estas se basan en la gramática que se elija, siendo diferente si el símbolo a analizar es terminal o no terminal.

Sumado a las directrices propias de la gramática se agregaron 3 niveles de validación extras, esto por si hay un símbolo que tuviera más de una producción, en estos casos se analiza la entrada y si fuera necesario las dos posiciones siguientes de la cadena posteriores a la entrada, buscando la igualdad entre la entrada y las posibles producciones, se hace la comparación hasta que solamente quede una opción, y por esta es la que se opta.

• Producción símbolo no terminal

ϵ , Lado izquierdo de la producción, Lado derecho de la producción

• Producción símbolo terminal

Simbolo terminal, Simbolo terminal, ϵ

Paradigmas utilizados

POO Se utilizó este paradigma para crear objetos de clase Gramatica y Produccion , para poder llevar de una forma más organizada el control de los datos provenientes del archivo de entrada.

Conclusiones

El uso de autómatas de pila es una excelente opción para hacer análisis sintáctico, tanto por su simplicidad como ya estar estandarizados los pasos a seguir para su implementación

Se debe incluir en la estimación del tiempo de programación, una buena parte de tiempo para corregir errores y solucionar imprevistos, ya que si no puede que el proyecto no se termine a tiempo.

.