

MANUAL TÉCNICO DE LA PROYECTO UNO “GRAMATICAS TIPO 2”

INDICE.....

- **INTRODUCCIÓN**.....
- **CLASES UTILIZADAS**.....
 - 1.**CLASE AUTOMATA**.....
 - 2.**CLASE GRAMATICA**.....
 - 3.**CLASE TRANSICION**
 - 4.**CLASE PRODUCCION**
- **PILA UTILIZADA**.....
- **CLASES UTILIZADAS**.....
- **MODELO**.....

INTRODUCCIÓN:

El presente programa consiste en una actualización del proyecto anterior, con la variación de que ahora este permite leer gramáticas tipo 2, creando su respectivo autómata de pila, incluyendo funciones nuevas como modificar una gramática ya creada, borrar producciones, crear reportes de cómo se ingresan datos a la pila.

El proyecto fue elaborado en Python, es una aplicación en consola que nos va indicando los datos a ingresar y nos devuelve los resultados necesarios ya sean en consola o como imágenes.

El programa funciona por medio de submenús en donde se va ingresando uno por uno para ir usando las diferentes funciones que este posee.

Para la elaboración del proyecto se utilizaron diversos paradigmas de programación como la funcional para la creación de las funciones de la pila y la orientada a objetos para la creación de autómatas de pila y gramáticas.

CLASES UTILIZADAS:

• **CLASE AUTOMATA:**

Esta clase, fue utilizada para la creación del autómata de pila, se creó de manera ordenada con distintos elementos pertenecientes al autómata de pila los cuales son la séxtupla que ellos poseen, de esta forma es más sencillo graficar el autómata y validar las cadenas con la pila de este.

Esta clase posee como atributos los atributos de un autómata de pila con diferentes tipos de datos que representen a los atributos del autómata y posee un método.

Los atributos de la clase autómata son los siguientes:

- Nombre de tipo string para guardar el nombre que tendrá el autómata este será el mismo que el de la gramática
- Alfabeto que es una lista de string para guardar el alfabeto del autómata que será el mismo que los no terminales de la gramática
- Estados es una lista de string que guarda los estados que tendrá el autómata de pila
- Estados de aceptación es una lista de tipo string que guarda los estados de finalización del autómata de pila
- Alfabeto de pila, es una lista de tipo string que guardara los elementos que se guardan en la pila
- Transiciones la cual es una lista de tipo transición una clase extra que se creó para guardar las transiciones del autómata de pila
- Estado inicial es una variable de tipo string para guardar el estado del que comenzara el autómata de pila

El método que se creó para esta clase fue un método toString() que nos devuelve un string con los datos de la séxtupla del autómata de pila.

• CLASE GRAMATICA

Esta clase fue creada para guardar los elementos que posee nuestra gramática, posee los elementos que tiene una gramática de tipo 2, los cuales son los no terminales, terminales, estado inicial y el conjunto de reglas de producciones.

Esta clase posee diferentes atributos para representar los elementos de la gramática de tipo dos, se utilizo programación orientada objetos para representar a esta, los elementos y atributos usados para esta gramática son:

- Nombre de tipo string el cual sirve para identificar a la gramática ya que esta puede ser modificada si se ingresa el mismo nombre
- Terminales, el cual es una lista de tipo string para guardar los terminales que tendrá la gramática
- No terminales, la cual es una lista de tipo string que nos servirá para guardar los no terminales que posee la gramática
- No terminal inicial de tipo string, el cual guarda en una variable el que será el no terminal inicial
- Producciones que es una lista de tipo producción la cual es una clase extra que se creó para representar las producciones que tendrá una gramática

La clase gramática también posee un método toString() que devuelve un string con la información detallada de los elementos que posee la gramática

• CLASE TRANSICION

Esta es una clase extra que se creó con programación orientada a objetos, la cual nos sirve para representar como será una transición de un estado a otro para nuestro autómata de pila generado, esta posee los mismos elementos de una transición escrita los cuales son, estado de origen, estado de destino, carácter leído, carácter sacado de pila y el carácter ingresado a la pila.

Para representar estos se crearon diferentes atributos en la clase para determinar y representar cada uno de estos elementos de una transición los cuales son:

- Estado de origen el cual es un string que representa el estado del que sale una transición
- Estado de destino el cual es un string que representa el estado al que se dirige la transición
- Carácter leído el representa el carácter con el cual se pasa a un estado al otro y está representado con un string
- Carácter sacado de pila el cual es un string y representa el elemento de la pila que se sacara cuando se lee ese carácter en esa transición
- Carácter ingresado en pila el cual es un string que representa un elemento a sacar de la pila

Esta clase posee un método toString el cual devuelve un string con la información de la transición que se creo

• CLASE PRODUCCION

Esta es una clase extra que se creó con programación orientada a objetos, la cual nos sirve para representar como será una producción de una gramática, esta posee los mismos elementos de una producción los cuales son, el estado del que sale la producción y una lista de expresiones que puede tener esta

Para representar estos se crearon diferentes atributos en la clase para determinar y representar cada uno de estos elementos de una producción los cuales son:

- Estado de partida el cual es un string que representa el estado del que salen las producciones
- Expresiones el cual es una lista de string que posee varias expresiones que representan las producciones que puede tomar el estado del que salen

Esta clase posee un método toString el cual devuelve un string con la información de la produccion que se creo

PILA UTILIZADA:

- **MÉTODO PUSH**

Este método fue creado para insertar al principio de una lista el elemento a guardar, tal como funciona una pila.

Este método consta de dos parámetros el primero es la lista la cual se ingresan los datos como una pila y el dato a guardar.

Para guardar el dato se utiliza el método `append()` de la lista como se usa normalmente en una lista pero la diferencia será el método `pop` que devolverá ese valor tal como funciona una pila que el ultimo en entrar es el primero en salir

- **MÉTODO POP**

Este método fue creado para devolver y borrar el último elemento de nuestra pila tal como funciona una pila.

Este tiene como parámetro la lista a la cual se quiere usar esta función, para ello se toma el último elemento de la lista se guarda en una variable luego se borra el último elemento de la lista con el método `clear` de una lista normal y se le pasa como parámetro la variable con el dato del último elemento por ultimo la función devuelve el valor de la variable la cual contiene el último elemento que se desea.

LIBRERIAS UTILIZADAS:

- Librería SYS
- Librería PATH
- Librería OS
- Librería TIME
- Librería GRAPHVIZ

MODELO:

