Universidad de San Carlos de Guatemala

Introducción a la Programación de Computación 2

Ing. Zulma

Aux. Luis Yela

Manual Tecnico

Elmer Gustavo Sánchez García

201801351

Fecha de entrega 3/03/2020

Menú Principal Gramática Tipo 2

```
def menuGramticaTiDos():
    while True:
        print("------------------Gramatica Tipo 2---------------------")
        print("|                                                     |")
        print("| 1.Ingresar/Modificar Gramatica                      |")
        print("| 2.Generar Automata de Pila                          |")
        print("| 3.Visualizar automata                               |")
        print("| 4.Validar Cadena                                    |")
        print("|   SALIR                                             |")
        print("|                                                     |")
        print("-------------------------------------------------------")
        print("")
        print(">> ",end="")
        opcion = input()
        opcion = opcion.strip()
        if opcion == "1":
            os.system("cls")
            menuNuevoGramaticaDos()
        elif opcion == "2":
            os.system("cls")
            nombreGramatica = setNombreGramticaDos()
            nombrePila = setNombreAutomataPila()
            if is_empty(nombreGramatica) == False and is_empty(nombrePila) == False:
                ManejadorAutomaPila.new_automataPila(nombreGramatica,nombrePila)
            else:
                ManejadorAutomaPila.alerta("Datos Vacios")
```

La función principal de este método es dar la opciones a escoger determinada función del programa como crear gramática evaluar cadenas, generar autómata de pila, visualizar autómata y validar cadena

Menú Gramática Tipo 2

```
def menuNuevoGramaticaDos():
    nombre = setNombreGramticaDos()
    if nombre != "update":
        if ManejadorGramaticaDos.newGramaticaDos(nombre) == True:
            while True:
                print("--------------Menu Gramatica Tipo 2--------------------")
                print("|                                                     |")
                print("| 1.Ingresar Terminales                               |")
                print("| 2.Ingresar No terminales                            |")
                print("| 3.Ingresar No Terminal Inicial                      |")
                print("| 4.Ingresar Producciones                             |")
                print("| 5.Borrar Produccion                                 |")
                print("|   SALIR                                             |")
                print("|                                                     |")
                print("-------------------------------------------------------")
                print("")
                print(">> ",end="")
                opcion = input()
                opcion = opcion.strip()
                if opcion == "1":
                    os.system("cls")
                    ManejadorGramaticaDos.setNewTerminal(nombre)
                elif opcion == "2":
                    os.system("cls")
                    ManejadorGramaticaDos.setNewNoTerminal(nombre)
                elif opcion == "3":
                    os.system("cls")
                    ManejadorGramaticaDos.setNTInicial(nombre)
                elif opcion == "4":
                    os.system("cls")
```

Este menu es el encargado de toda la gramtica como crear terminales, no terminales, no terminal inicial ,crear producciones y eliminar producciones.

Alerta

```
def alerta(mensaje):
    os.system("cls")
    print("------------------------ALERTA-------------------------")
    print("|                                                     |")
    print("| Error:                                              |")
    print("| ->>No Se Realizo La Operacion                       |")
    print(f"| -->>{mensaje}                ")
    print("|                                                     |")
    print("-------------------------------------------------------")
```

Este son notificaciones por alguno posible error presentado en la ejecución del programa o cuando se quiere notificar alguna operación realizada.

Get_RutaAutomataDePila

```
def get_rutaAutomataPila(nombre):
    nombre = nombre.strip()
    automata = get_AutomataGramtica(nombre)
    if automata != None:
        root = Tk()
        root.filename =  filedialog.asksaveasfilename(
initialdir = "/",title = "Select CSV file",filetypes
 = (("CSV files","*.csv"),("all files","*.*")))
        ruta = ""
        ruta = root.filename
        comando  = ""
        if is_empty(ruta.strip()) == False:
            ruta_csv = f"{ruta}.csv"
            root.destroy()
            generar_CSVAutomata(nombre,ruta_csv)
        else:
            alerta("No escribio ningun nombre")
            root.destroy()
            return None
    else:
        alerta("No se encontro el automata ): ")
        return None
```

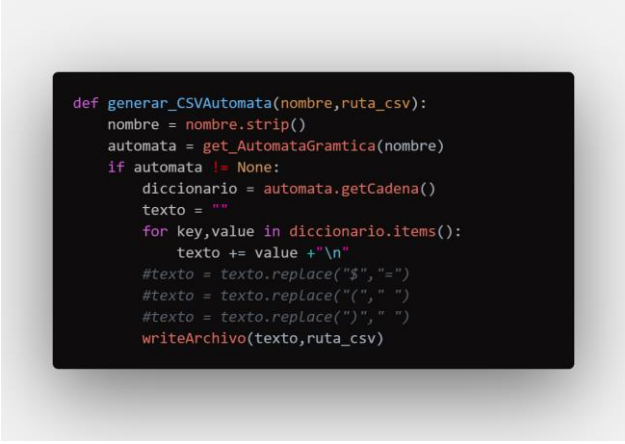
Este método se encarga de obtener la ruta dependiendo si es para una gramática o un afd, se encarga de forma mas visual, y mas sencilla de obtener la ruta.

Nuevo Autómata Pila

```
def new_automataPila(nombreGramatica,nombreAutomata):
    nombreGramatica = nombreGramatica.strip()
    nombreAutomata = nombreAutomata.strip()
    gramatica = ManejadorGramaticaDos.getObjeto
(nombreGramatica)
    if gramatica != None:
        if duplicate_data(nombreAutomata) == False:
            nuevoAutomata = AutomataPila
(nombreAutomata,[],[],"","",{},[],[],""
,nombreGramatica,{})
            listaAutomataPila.append(nuevoAutomata)
            set_datosAutomata
(nombreAutomata,nombreGramatica)
        else:
            alerta("Datos Duplicados :v")
            return False
```

Este método es el encargado de crear un nuevo autómata de pila, cuando se le envía el nombre se genera un autómata de pila con forme se va ejecutado el código, empieza llenarse el autómata.

Escribir en archivo CSV



```
def generar_CSVAutomata(nombre,ruta_csv):
    nombre = nombre.strip()
    automata = get_AutomataGramtica(nombre)
    if automata != None:
        diccionario = automata.getCadena()
        texto = ""
        for key,value in diccionario.items():
            texto += value +"\n"
        #texto = texto.replace("$","=")
        #texto = texto.replace("("," ")
        #texto = texto.replace(")"," ")
        writeArchivo(texto,ruta_csv)
```

Este método se encarga de traer todos los datos unirlos y enviarse al metodo write archivo para escribir el archivo csv, le envia la ruta y el texto.

Transiciones

```
def get_TrancisionPila(automata,clave,buscado):
    diccionario = automata.getTrancisiones()
    for key,value in diccionario.items():
        if key == clave:
            lista = value
            for cadena in lista:
                cadena = cadena.strip()
                if cadena.count(buscado) == 1:
                    nuevaCadena = cadena
                    nuevaCadena = nuevaCadena.strip()
                    cadena = cadena.split(";")
                    paretA = cadena[0]
                    parteB = cadena[1]
                    parteB = parteB.split(",")
                    estado = parteB[0]
                    insePila = parteB[1]
                    insePila = insePila.strip()
                    return estado,insePila,nuevaCadena
    return None,None,None
```

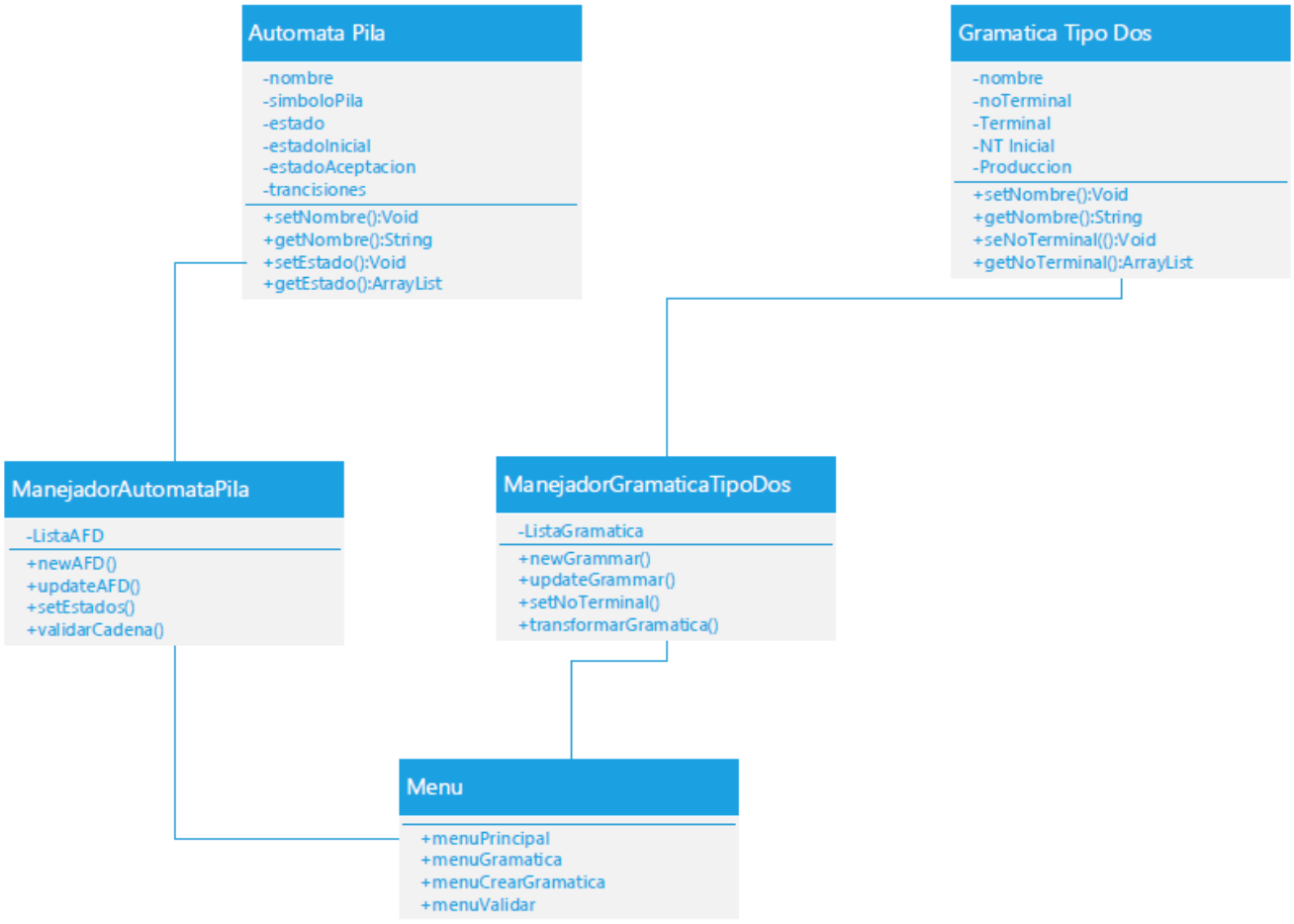
Este método se encarga de generar retornar lo que se va insertar, a la hora de realizar una transición, para el método principal que es el de validar cadena.

Insertar en Pila

```
def insert_Pila(pila,cadena):
    try:
        cadena = cadena.split(" ")
        for valor in reversed(cadena):
            #print(valor)
            pila.append(valor)
        return pila
    except IndexError as e:
        alerta(e)
        return None
```

Este método se encarga de insertar en pila , pero de forma inversa por la forma en esta guardado los datos en la lista.

UML


Automata Pila
-nombre
-simboloPila
-estado
-estadoInicial
-estadoAceptacion
-trancisiones
+setNombre():Void
+getNombre():String
+setEstado():Void
+getEstado():ArrayList
Gramatica Tipo Dos
-nombre
-noTerminal
-Terminal
-NT Inicial
-Produccion
+setNombre():Void
+getNombre():String
+seNoTerminal():Void
+getNoTerminal():ArrayList
ManejadorAutomataPila
-ListaAFD
+newAFD()
+updateAFD()
+setEstados()
+validarCadena()
ManejadorGramaticaTipoDos
-ListaGramatica
+newGrammar()
+updateGrammar()
+setNoTerminal()
+transformarGramatica()
Menu
+menuPrincipal
+menuGramatica
+menuCrearGramatica
+menuValidar