

ANDY ROBERTO JIMENEZ MACNAB

202111490

Laboratorio de Lenguajes Formales y de Programación

Sección A+

Manual de Técnico

Autómata finito determinista

y Autómata de pila

28/10/2022

Universidad de San Carlos de Guatemala

Proyecto2

MANUAL TECNICO:

Principio de Programación:

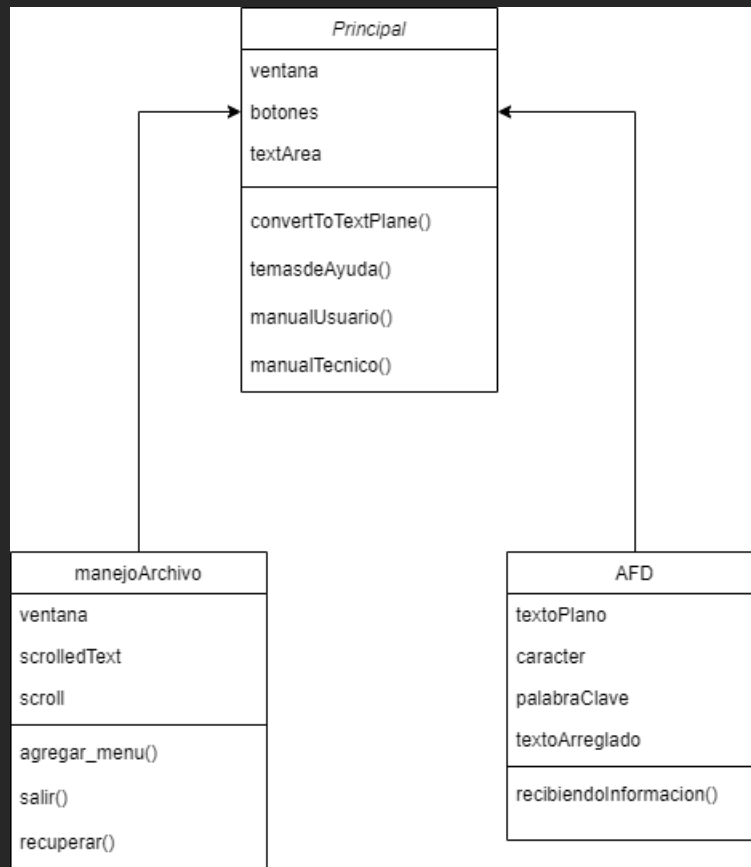
- Programación orientada a objetos:

En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite **separar los diferentes componentes de un programa**, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promueve la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

Convenciones de Nomenclatura:

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: *MiBlogDeDesarrollo*.

Diagrama de Clases:



Métodos principales:

1. `convertToTextPlane()`:

Este método se encarga de convertir el archivo GPW ingresado por el usuario y posteriormente cargado en el `textArea` de tkinter en un texto plano para mayor facilidad para el programa al momento de analizarlo.

```
def convertToTextPlane(self):
    self.textoPlano=str(self.textArea.get("1.0","end-1c"))
    print(self.textoPlano)
    print("-----")
    automata.recibiendoInformacion(self.textoPlano)
```

2. recibiendoInformacion():

```
def recibiendoInformacion(textoPlano):
    errores=0
    textoFinal=[]

    palabraClave=""
    listaFilas=textoPlano.split("\n")
    print(listaFilas)
    for elemento in listaFilas:
        palabraClave=""
        if elemento[0]!="<":
            errores+=1
            print(errores)

        for caracter in elemento:

            if caracter == '<':
                continue
            if caracter == '>':
                continue
            else:
                palabraClave+= caracter

        if palabraClave == "Operacion":

            listaOperacion=[]

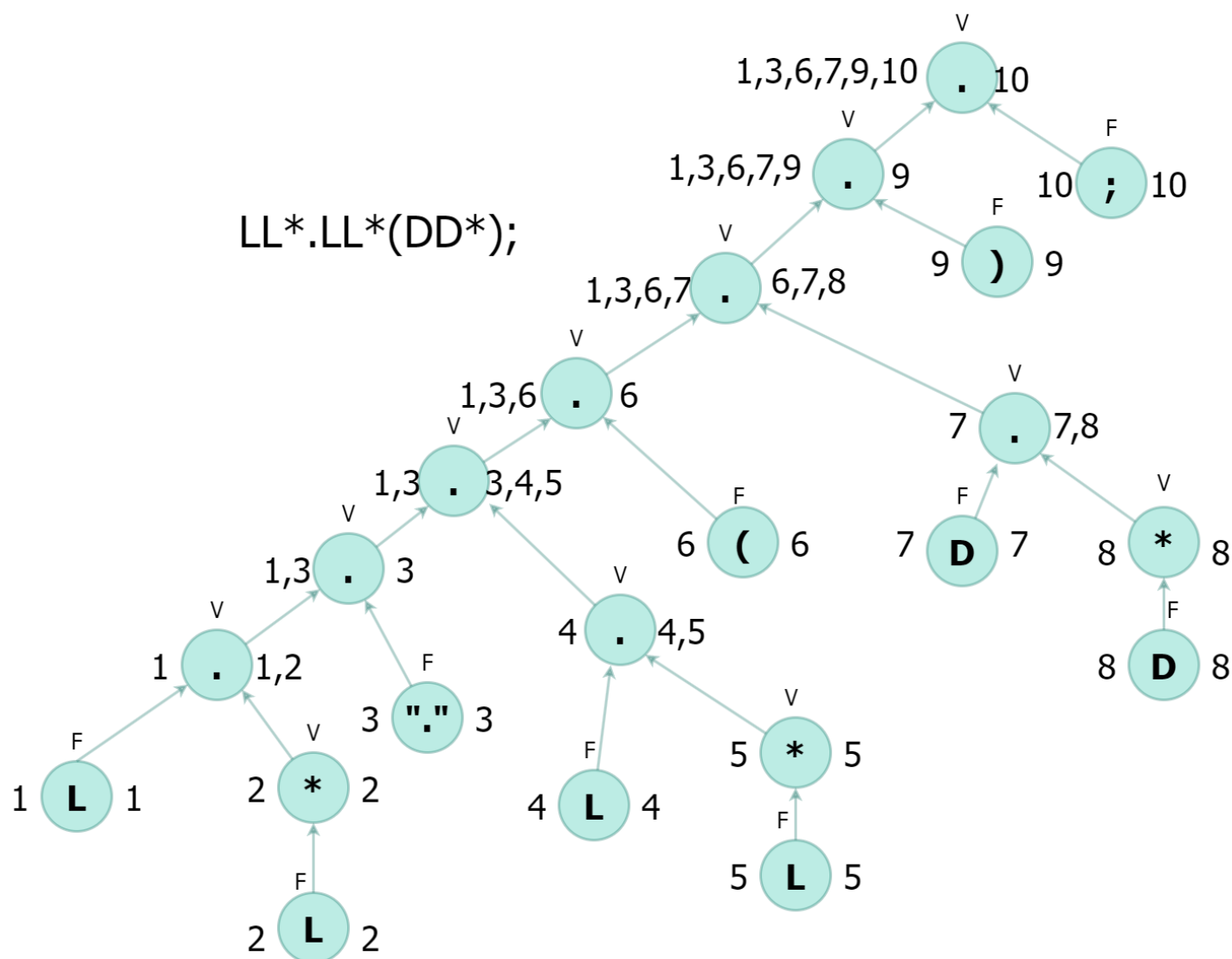
            operacion=elemento.split(" ")
```

Esta función podría considerar la función principal de todo el programa ya que es el método que por medio del texto plano proporcionado comienza a ejecutar el autómata finito determinista con el fin de poder hallar la solución matemática a los distintos problemas contenidos en el archivo, finalmente luego de ser analizado, genera un archivo HTML y CSS con todos los datos estructurados.

Expresión Regular	Token
<!--Controles	Inicio de controles
Contenedor ID	Componente tipo contenedor
Boton ID	Componente tipo botón
Clave ID	Componente tipo clave
Etiqueta ID	Componente tipo etiqueta
Texto ID	Componente tipo texto
Clave ID	Componente tipo clave
Controles-->	Fin de sección controles
<!--propiedades	Inicio de las propiedades
//#\$inicio de ID	Inicio de propiedades por ID
//#\$fin de ID	Fin de propiedades por ID
.setAlto	El alto del componente
.setAncho	El ancho del componente
setColorFondo	El color de fondo contenedor
.setColorLetra	Color de letra
Propiedades-->	Fin de las propiedades
<!--Colocacion	Inicio de la colocación
.setPosicion	Posición del componente
.add	Añadir el componente al html
Colocación-->	Fin de la colocacion

AUTOMATA FINITO DETERMINISTA POR METODO DEL ARBOL:

LL*.LL*(DD*);



I	SIG(I)
1 L	2,3
2 L	2,3
3 .	4,5,6
4 L	5,6
5 L	5,6
6 (7,8,9
7 D	8,9
8 D	8,9
9)	10
10 ;	

$S_0 = \{1, 3, 6, 7, 9, 10\}$

L.(D):

$Sig(1) = \{2, 3\} = S_1$

$Sig(2) = \{2, 3\} = S_1$

$Sig(3) = \{4, 5, 6\} = S_2$

$Sig(3) = \{4, 5, 6\} = S_2$

$Sig(4) = \{5, 6\} = S_2$

$Sig(5) = \{5, 6\} = S_2$

$Sig(6) = \{7, 8, 9\} = S_3$

$\text{Sig}(6)=\{7,8,9\}=S3$

$\text{Sig}(7)=\{8,9\}=S3$

$\text{Sig}(8)=\{8,9\}=S3$

$\text{Sig}(9)=\{10\}=S4$

$\text{Sig}(7)=\{8,9\}=S3$

$\text{Sig}(9)=\{10\}=S4$

