



Instituto Politécnico Nacional Escuela Superior de Cómputo

Teoría de la computación

Practica 6: Autómata De Pila
Profesor: Juarez Martinez Genaro
4CM6

Torres Abonce Luis Miguel
Mayo 2022

Contents

1	Introducción	3
2	Marco Teórico	4
3	Desarrollo	5
3.1	Código	5
3.2	Explicación	10
3.3	Capturas de ejecución	11
4	Conclusiones	13
5	Bibliografía	14
6	Implementación latex	15

1 Introducción

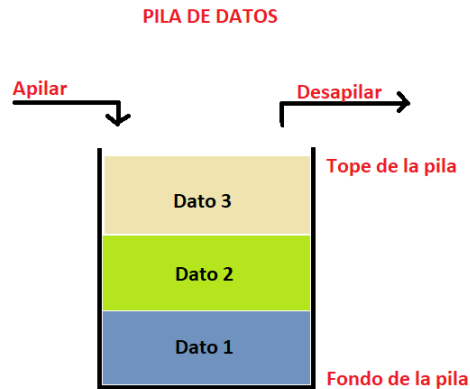
Descripción del problema:

El problema consiste en programar el autómata de pila para que reconozca el lenguaje libre de contexto: $0^n 1^n \mid n \geq 1$.

El programa deberá de dar la opción al usuario si quiere que la cadena se genere automáticamente (no mas de 100,000 caracteres) o introducir la cadena de manera manual; además se tendrá que mandar a un archivo de texto las descripciones instantáneas (IDs); adicionalmente se deberán de graficar las cadenas que sean menores a 10 de tamaño

2 Marco Teórico

Pila. Son colecciones de elementos ordenados que únicamente permiten dos acciones: Añadir un elemento a la pila. Sacar un elemento de la pila. La peculiaridad es que el último elemento en entrar es el primero en salir. En inglés se conocen como estructuras LIFO (Last In First Out).



Los lenguajes libres del contexto son aquellos que pueden ser reconocidos por un autómata de pila determinístico o no determinístico. Como toda gramática se definen mediante una cuádrupla $G = (N, T, P, S)$, siendo

- N es un conjunto finito de símbolos no terminales
- T es un conjunto finito de símbolos terminales $N \cap T = \emptyset$
- P es un conjunto finito de producciones
- S es el símbolo distinguido o axioma

3 Desarrollo

3.1 Código

```
import random
import sys, os
from time import sleep
import turtle
def animacion(ES,numeroX):
    if numeroX==0:
        if ES==1:#Entra una X
            X10.goto(0,170)
            X10.shape(rutaX1)
            X10.goto(0,-315)
    elif numeroX==1:
        if ES==1:#Entra una X
            X9.goto(0,170)
            X9.shape(rutaX2)
            X9.goto(0,-265)
        else:#Sale una X
            X10.goto(0,170)
            X10.hideturtle()
    elif numeroX==2:
        if ES==1:
            X8.goto(0,170)
            X8.shape(rutaX2)
            X8.goto(0,-215)
        else:
            X9.goto(0,170)
            X9.hideturtle()
    elif numeroX==3:
        if ES==1:
            X7.goto(0,170)
            X7.shape(rutaX2)
            X7.goto(0,-165)
        else:
            X8.goto(0,170)
            X8.hideturtle()
    elif numeroX==4:
        if ES==1:
            X6.goto(0,170)
            X6.shape(rutaX2)
            X6.goto(0,-115)
        else:
            X7.goto(0,170)
            X7.hideturtle()
    elif numeroX==5:
        if ES==1:
            X5.goto(0,170)
            X5.shape(rutaX2)
            X5.goto(0,-65)
        else:
            X6.goto(0,170)
            X6.hideturtle()
    if numeroX==6:
        if ES==1:
```

```

        X4.goto(0,170)
        X4.shape(rutaX2)
        X4.goto(0,-15)
    else:
        X5.goto(0,170)
        X5.hideturtle()
elif numeroX==7:
    if ES==1:#Entra una X
        X3.goto(0,170)
        X3.shape(rutaX2)
        X3.goto(0,35)
    else:#Sale una X
        X4.goto(0,170)
        X4.hideturtle()
elif numeroX==8:
    if ES==1:
        X2.goto(0,170)
        X2.shape(rutaX2)
        X2.goto(0,85)
    else:
        X3.goto(0,170)
        X3.hideturtle()
elif numeroX==9:
    if ES==1:
        X1.goto(0,170)
        X1.shape(rutaX2)
        X1.goto(0,135)
    else:
        X2.goto(0,170)
        X2.hideturtle()
elif numeroX==10:
    X1.goto(0,170)
    X1.hideturtle()

doc=open("ids.txt","w")#abrir archivo
print("Menu:")
print("Ingresar cadena.")
print("1. Manual.")
print("2. Aleatoriamente")
eleccion=int(input("Elige una opcion: "))
w=0
if eleccion==1:
    cadena=input("Introduce la cadena: ")
    for i in cadena:
        w=w+1#tamano de la cadena
else:
    cadena=''
    x=random.randint(1,100)#Longitud 1-100,000
    y=0
    while(y!=x):
        cadena=cadena+"0"#Juntar la cadena anterior para formar la nueva
        y+=1
    z=random.randint(1,100)
    y=0
    while(y!=z):
        cadena=cadena+"1"#Juntar la cadena anterior para formar la nueva
        y+=1
    w=x+z

```

```

if w>10: animar=1 #comprbar animacion
else: animar=0
def rutaImagen(nombreImagen):#recibe 1 argumento
    pathScript=sys.argv[0]#ruta de este codigo
    carpeta,script=os.path.split(pathScript)#ruta de la carpeta
    pathFondo=os.path.join(carpeta, nombreImagen)#0btiene la ruta absoluta
    return pathFondo#regresa la ruta
#iniciamos la animacion
if animar==0:
    ventana = turtle.Screen()
    ventana.setup(1350,950)#definir dimension de ventanas
    rutaX1=rutaImagen("X.gif")
    rutaX2=rutaImagen("X.gif")
    rutaX3=rutaImagen("X.gif")
    rutaX4=rutaImagen("X.gif")
    rutaX5=rutaImagen("X.gif")
    rutaX6=rutaImagen("X.gif")
    rutaX7=rutaImagen("X.gif")
    rutaX8=rutaImagen("X.gif")
    rutaX9=rutaImagen("X.gif")
    rutaX10=rutaImagen("X.gif")
    ventana.addshape(rutaX1)
    ventana.addshape(rutaX2)
    ventana.addshape(rutaX3)
    ventana.addshape(rutaX4)
    ventana.addshape(rutaX5)
    ventana.addshape(rutaX6)
    ventana.addshape(rutaX7)
    ventana.addshape(rutaX8)
    ventana.addshape(rutaX9)
    ventana.addshape(rutaX10)
    X1=turtle.Turtle()
    X2=turtle.Turtle()
    X3=turtle.Turtle()
    X4=turtle.Turtle()
    X5=turtle.Turtle()
    X6=turtle.Turtle()
    X7=turtle.Turtle()
    X8=turtle.Turtle()
    X9=turtle.Turtle()
    X10=turtle.Turtle()
    X1.speed(1)
    X1.penup()
    X2.speed(1)
    X2.penup()
    X3.speed(1)
    X3.penup()
    X4.speed(1)
    X4.penup()
    X5.speed(1)
    X5.penup()
    X6.speed(1)
    X6.penup()
    X7.speed(1)
    X7.penup()
    X8.speed(1)
    X8.penup()
    X9.speed(1)

```

```

X9.penup()
X10.speed(1)
X10.penup()
tortuCuad=turtle.Turtle()
tortuCuad.speed(0)
tortuCuad.penup()
tortuCuad.color("navy")
tortuCuad.begin_fill()
tortuCuad.goto(-150,-340)
tortuCuad.goto(150,-340)
tortuCuad.goto(150,150)
tortuCuad.goto(-150,150)
tortuCuad.goto(-150,-340)
tortuCuad.end_fill()
tortuCuad.goto(-500,0)
tortuCuad.write("Cadena:"+cadena,font=("courier",20, "bold"))
tortuCuad.left(90)
p=-381
tortuCuad.goto(p,0)
pila=[]#declaramos la pila
y=0
i=0
exis=0
alfa="Z0"
doc.write("q"+" "+cadena+" "+alfa+"\n")
while(y!=w):
    if animar==0: p+=16
    if cadena[y] == "0":#estado q
        pila.append("x")#agregar X
        alfa="X"+alfa
        doc.write("q"+" "+cadena[y:]+ " "+alfa+"\n")
        if animar==0:#la cadena es menor a 10 entonces se grafica
            animacion(1,exis)
            tortuCuad.goto(p,0)
            exis+=1
    elif cadena[y] == "1":#estado p
        try:
            pila.pop()#Quitar X
            doc.write("p"+" "+cadena[y:]+ " "+alfa[i:]+ "\n")
            if animar==0:
                animacion(0,exis)
                tortuCuad.goto(p,0)
                exis-=1
            i+=1
        except:
            doc.write("f"+" "+cadena[y-1:]+ " "+alfa[i-1:]+ "\n")
            if animar==0:
                tortuCuad.goto(200,0)
                tortuCuad.write("Cadena no Aceptada \nhay mas 1s que 0s",font=("courier",20, "bold"))
                sleep(5)
            doc.close()
            exit(1)
    y+=1
if not pila:#estado f
    print("Ta vacia")#estado f aceptado
    doc.write("f"+" "+vacio+" "+alfa[i:]+ "\n")
    if animar==0:
        tortuCuad.goto(200,0)

```



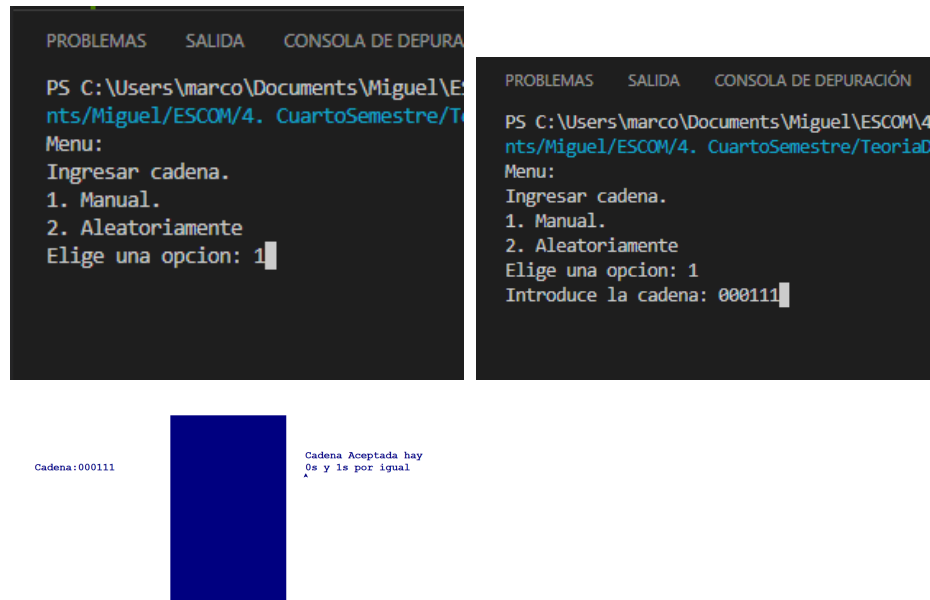
```
        tortuCuad.write("Cadena Aceptada hay \n0s y 1s por igual",font=("courier",20, "bold"))
        sleep(5)
    else:
        doc.write("f"+"", "+cadena[y-1:]"+"", "+alfa[i-1:]"+"\\n")
        if animar==0:
            tortuCuad.goto(200,0)
            tortuCuad.write("Cadena no Aceptada hay \nmas 0s que 1s",font=("courier",20, "bold"))
            sleep(5)
doc.close()
```

3.2 Explicación

Este programa se desarrollo en el lenguaje python; primeramente define todas las librerías a utilizar, después tengo mi primera función que es animación la cual como su nombre dice es para animar el autómata de pila la cual recibe 2 parámetros ES que significa si entra o sale una x del autómata y el numero de X que hay en la pila, si ,por ejemplo, hay 2 X en la pila y queremos meter otra X lo que hacemos es mostrar la 8va X y llevarla al lugar que corresponde, de caso contrario la 9va X es la que saldrá de la pila y desaparecerá; posteriormente abro el documento donde guardar las IDs y despliego el menú y se determina si la cadena se generara aleatoriamente o si se introducirá de manera manual, después tengo una función para encontrar la ruta absoluta de las imágenes X para gráfica.Después definimos el tamaño de la venta a graficar y se agregar todas las rutas y se agregan a la ventana, después dibujo un rectángulo que representara la pila, declaramos la pila como `pila=[]` para meter datos a la pila se utiliza `pila.append(dato)` para sacar un dato de la pila se utiliza `pila.pop()`, después tenemos un ciclo que recorrerá toda la cadena tenemos un if que representara al estado q donde si se encuentra un 0 en la cadena agregara una X a la pila e imprimirá en el documento la descripción instantánea.En caso de que el carácter en la cadena sea un 1 entonces se quitara una X de la pila y se escribirá la descripción instantánea en el documento; se utiliza un try except para cuando no se pueda quitar una X (ya no hay en la pila) Entonces la cadena no es aceptada y se termina el programa además de escribirlo en en el documento su IDs, después fuera del ciclo si la pila esta vacía se indica que se acepto la cadena, caso contrario significa que hay mas 0 que 1 por lo tanto la cadena no es aceptada y se cierra la documento. Esto seria básicamente la función del programa.

3.3 Capturas de ejecución

Manual:



4 Conclusiones

El programa fue bastante sencillo de resolver pero como siempre la parte de la animación es la mas tardada mas no complicada; la parte de crear la pila creí que seria mas difícil pero cuando me puse a investigar, leí que las "Listas" que usaba en python ya era una pila e incluso tenia el comando pop que nunca había utilizado, gracias a esto fue mas sencillo de resolver la problemática

5 Bibliografía

Hopcroft, J. E. (1979). Introduction to Automata Theory, Languages, and Computation [Paperback] [Jan 01, 1979] John E. Hopcroft (2.a ed., Vol. 2). Pearson. Pila. (s. f.). [Ilustración]. programacionpython. <https://programacionpython80889555.files.wordpress.com/2018/08/programacionpython-2a-edicion-john-e-hopcroft.pdf>

6 Implementación latex

```
\documentclass[12pt,letterpaper]{article}
\usepackage[utf8]{inputenc}
\usepackage{graphicx}
\usepackage{changepage}
\usepackage{float}

\usepackage[left=1.5cm, right=1.5cm]{geometry}

\begin{document}

\pagestyle{plain}{
\pagestyle{empty}
\changepage{3cm}{1cm}{-0.5cm}{-0.5cm}{-2cm}{-2cm}{}{}
\noindent

{\small
\begin{tabular}{p{0.626\textwidth} p{0.50\textwidth} }
\includegraphics[width=5cm]{logo.jpg} & \includegraphics[width=5cm]{logoESCOM.jpg}
\end{tabular}
}
\begin{center}
\par\vspace{2cm}
{
\Huge\textbf{
Instituto Politécnico Nacional\\
Escuela Superior de Cómputo \\[1cm]
Teoría de la computación \\[1cm]
}
}
\par\vspace{0.5cm}
{
\Large\textbf{Practica 6: Autómata De Pila \ \ Profesor: Juarez Martinez Genaro \\[.5cm] 4CM6}
}
\par\vspace{1cm}
{
\large\textbf{Torres Abonce Luis Miguel \ \ Mayo 2022}
}
\par\vspace{3cm}
\end{center}
\clearpage
\newpage \tableofcontents\
}
\newpage
\section{Introducción}
\large\textwidth{
Descripción del problema: \\[.5cm]
\justify
El problema consiste en programar el autómata de pila para que reconozca el lenguaje libre de contexto:  $0^n 1^n | n \geq 1$ .
\ \ El programa deberá de dar la opción al usuario si quiere que la cadena se genere automáticamente (no mas de 100,000
caracteres) o introducir la cadena de manera manual; además se tendrá que mandar a un archivo de texto las
descripciones instantáneas (IDs); adicionalmente se deberán de graficar las cadenas que sean menores a 10 de tamaño
}
\newpage
\section{Marco Teórico}
```

```

\large\textwidth\justify{
  Pila. Son colecciones de elementos ordenados que únicamente permiten dos acciones: Añadir un elemento a la pila.
  Sacar un elemento de la pila. La peculiaridad es que el último elemento en entrar es el primero en salir. En
  inglés se conocen como estructuras LIFO (Last In First Out).\includegraphics[width=8cm]{pila.png}\\
  Los lenguajes libres del contexto son aquellos que pueden ser reconocidos por un autómata de pila determinístico o
  no determinístico. Como toda gramática se definen mediante una cuadrupla  $G = (N, T, P, S)$ , siendo \\
  - N es un conjunto finito de símbolos no terminales\\
  - T es un conjunto finito de símbolos terminales  $N \cap T = \emptyset$ 
  - P es un conjunto finito de producciones\\
  - S es el símbolo distinguido o axioma
}
\newpage
\section{Desarrollo}
\subsection{Código}
\scriptsize{
\begin{verbatim}
import random
import sys, os
from time import sleep
import turtle
def animacion(ES,numeroX):
    if numeroX==0:
        if ES==1:#Entra una X
            X10.goto(0,170)
            X10.shape(rutaX1)
            X10.goto(0,-315)
        elif numeroX==1:
            if ES==1:#Entra una X
                X9.goto(0,170)
                X9.shape(rutaX2)
                X9.goto(0,-265)
            else:#Sale una X
                X10.goto(0,170)
                X10.hideturtle()
        elif numeroX==2:
            if ES==1:
                X8.goto(0,170)
                X8.shape(rutaX2)
                X8.goto(0,-215)
            else:
                X9.goto(0,170)
                X9.hideturtle()
        elif numeroX==3:
            if ES==1:
                X7.goto(0,170)
                X7.shape(rutaX2)
                X7.goto(0,-165)
            else:
                X8.goto(0,170)
                X8.hideturtle()
        elif numeroX==4:
            if ES==1:
                X6.goto(0,170)
                X6.shape(rutaX2)
                X6.goto(0,-115)
            else:
                X7.goto(0,170)
                X7.hideturtle()

```



```

elif numeroX==5:
    if ES==1:
        X5.goto(0,170)
        X5.shape(rutaX2)
        X5.goto(0,-65)
    else:
        X6.goto(0,170)
        X6.hideturtle()
if numeroX==6:
    if ES==1:
        X4.goto(0,170)
        X4.shape(rutaX2)
        X4.goto(0,-15)
    else:
        X5.goto(0,170)
        X5.hideturtle()
elif numeroX==7:
    if ES==1:#Entra una X
        X3.goto(0,170)
        X3.shape(rutaX2)
        X3.goto(0,35)
    else:#Sale una X
        X4.goto(0,170)
        X4.hideturtle()
elif numeroX==8:
    if ES==1:
        X2.goto(0,170)
        X2.shape(rutaX2)
        X2.goto(0,85)
    else:
        X3.goto(0,170)
        X3.hideturtle()
elif numeroX==9:
    if ES==1:
        X1.goto(0,170)
        X1.shape(rutaX2)
        X1.goto(0,135)
    else:
        X2.goto(0,170)
        X2.hideturtle()
elif numeroX==10:
    X1.goto(0,170)
    X1.hideturtle()

doc=open("ids.txt","w")#abrir archivo
print("Menu:")
print("Ingresar cadena.")
print("1. Manual.")
print("2. Aleatoriamente")
eleccion=int(input("Elige una opcion: "))
w=0
if eleccion==1:
    cadena=input("Introduce la cadena: ")
    for i in cadena:
        w=w+1#tamano de la cadena
else:
    cadena=''
    x=random.randint(1,100)#Longitud 1-100,000

```

```

y=0
while(y!=x):
    cadena=cadena+"0"#Juntar la cadena anterior para formar la nueva
    y+=1
z=random.randint(1,100)
y=0
while(y!=z):
    cadena=cadena+"1"#Juntar la cadena anterior para formar la nueva
    y+=1
w=x+z
if w>10: animar=1 #comprbar animacion
else: animar=0
def rutaImagen(nombreImagen):#recibe 1 argumento
    pathScript=sys.argv[0]#ruta de este codigo
    carpeta,script=os.path.split(pathScript)#ruta de la carpeta
    pathFondo=os.path.join(carpeta, nombreImagen)#Obtiene la ruta absoluta
    return pathFondo#regresa la ruta
#iniciamos la animacion
if animar==0:
    ventana = turtle.Screen()
    ventana.setup(1350,950)#definir dimension de ventanas
    rutaX1=rutaImagen("X.gif")
    rutaX2=rutaImagen("X.gif")
    rutaX3=rutaImagen("X.gif")
    rutaX4=rutaImagen("X.gif")
    rutaX5=rutaImagen("X.gif")
    rutaX6=rutaImagen("X.gif")
    rutaX7=rutaImagen("X.gif")
    rutaX8=rutaImagen("X.gif")
    rutaX9=rutaImagen("X.gif")
    rutaX10=rutaImagen("X.gif")
    ventana.addshape(rutaX1)
    ventana.addshape(rutaX2)
    ventana.addshape(rutaX3)
    ventana.addshape(rutaX4)
    ventana.addshape(rutaX5)
    ventana.addshape(rutaX6)
    ventana.addshape(rutaX7)
    ventana.addshape(rutaX8)
    ventana.addshape(rutaX9)
    ventana.addshape(rutaX10)
    X1=turtle.Turtle()
    X2=turtle.Turtle()
    X3=turtle.Turtle()
    X4=turtle.Turtle()
    X5=turtle.Turtle()
    X6=turtle.Turtle()
    X7=turtle.Turtle()
    X8=turtle.Turtle()
    X9=turtle.Turtle()
    X10=turtle.Turtle()
    X1.speed(1)
    X1.penup()
    X2.speed(1)
    X2.penup()
    X3.speed(1)
    X3.penup()
    X4.speed(1)

```

```

X4.penup()
X5.speed(1)
X5.penup()
X6.speed(1)
X6.penup()
X7.speed(1)
X7.penup()
X8.speed(1)
X8.penup()
X9.speed(1)
X9.penup()
X10.speed(1)
X10.penup()
tortuCuad=turtle.Turtle()
tortuCuad.speed(0)
tortuCuad.penup()
tortuCuad.color("navy")
tortuCuad.begin_fill()
tortuCuad.goto(-150,-340)
tortuCuad.goto(150,-340)
tortuCuad.goto(150,150)
tortuCuad.goto(-150,150)
tortuCuad.goto(-150,-340)
tortuCuad.end_fill()
tortuCuad.goto(-500,0)
tortuCuad.write("Cadena:"+cadena,font=("courier",20, "bold"))
tortuCuad.left(90)
p=-381
tortuCuad.goto(p,0)
pila=[]#declaramos la pila
y=0
i=0
exis=0
alfa="Z0"
doc.write("q"+" "+cadena+" "+alfa+"\n")
while(y!=w):
    if animar==0: p+=16
    if cadena[y] == "0":#estado q
        pila.append("x")#agregar X
        alfa="X"+alfa
        doc.write("q"+" "+cadena[y:]+" "+alfa+"\n")
        if animar==0:#la cadena es menor a 10 entonces se grafica
            animacion(1,exis)
            tortuCuad.goto(p,0)
            exis+=1
    elif cadena[y] == "1":#estado p
        try:
            pila.pop()#Quitar X
            doc.write("p"+" "+cadena[y:]+" "+alfa[i:]+"\n")
            if animar==0:
                animacion(0,exis)
                tortuCuad.goto(p,0)
                exis-=1
            i+=1
        except:
            doc.write("f"+" "+cadena[y-1:]+" "+alfa[i-1:]+"\n")
            if animar==0:
                tortuCuad.goto(200,0)

```

```

        tortuCuad.write("Cadena no Aceptada \nhay mas 1s que 0s",font=("courier",20, "bold"))
        sleep(5)
    doc.close()
    exit(1)

    y+=1
if not pila:#estado f
    print("Ta vacia")#estado f aceptado
    doc.write("f"+" "+"vacio"+" "+alfa[i:]+\n")
    if animar==0:
        tortuCuad.goto(200,0)
        tortuCuad.write("Cadena Aceptada hay \n0s y 1s por igual",font=("courier",20, "bold"))
        sleep(5)
else:
    doc.write("f"+" "+cadena[y-1:]+\n")
    if animar==0:
        tortuCuad.goto(200,0)
        tortuCuad.write("Cadena no Aceptada hay \nmas 0s que 1s",font=("courier",20, "bold"))
        sleep(5)
doc.close()
\newpage\subsection{Explicación}
\textwidth\large{
Este programa se desarrollo en el lenguaje python; primeramente define todas las librerías a utilizar, después tengo mi
primera función que es animación la cual como su nombre dice es para animar el autómata de pila la cual recibe 2
parámetros ES que significa si entra o sale una x del autómata y el numero de X que hay en la pila, si ,por ejemplo, hay 2
X en la pila y queremos meter otra X lo que hacemos es mostrar la 8va X y llevarla al lugar que corresponde, de caso
contrario la 9va X es la que saldrá de la pila y desaparecerá; posteriormente abro el documento donde guardar las IDs y
despliego el menú y se determina si la cadena se generara aleatoriamente o si se introducirá de manera manual, después
tengo una función para encontrar la ruta absoluta de las imágenes X para gráfica.Después definimos el tamaño de la venta a
graficar y se agregar todas las rutas y se agregan a la ventana, después dibujo un rectángulo que representara la pila,
declaramos la pila como pila=[] para meter datos a la pila se utiliza pila.append(dato) para sacar un dato de la pila se
utiliza pila.pop(), después tenemos un ciclo que recorrerá toda la cadena tenemos un if que representara al estado q donde
si se encuentra un 0 en la cadena agregara una X a la pila e imprimirá en el documento la descripcion instantánea.En caso
de que el carácter en la cadena sea un 1 entonces se quitara una X de la pila y se escribirá la descripción instantánea en
el documento; se utiliza un try except para cuando no se pueda quitar una X (ya no hay en la pila) Entonces la cadena no
es aceptada y se termina el programa además de escribirlo en en el documento su IDs, después fuera del ciclo si la pila
esta vacía se indica que se acepto la cadena, caso contrario significa que hay mas 0 que 1 por lo tanto la cadena no es
aceptada y se cierra la documento. Esto seria básicamente la función del programa.
}
\newpage
\subsection{Capturas de ejecución}
\textwidth
Manual:\\[.2cm]
\includegraphics[width=6cm]{man1.png}
\includegraphics[width=6cm]{man2.png}\\
\includegraphics[width=6cm]{man3.png}\\
\newpage Automatico:\\[.2cm]
\includegraphics[width=6cm]{aut1.png}\\
\includegraphics[width=8cm]{aut2.png}\\

\newpage\section{Conclusiones}
El programa fue bastante sencillo de resolver pero como siempre la parte de la animacion es la mas tardada mas no
complicada; la parte de crear la pila crei que seria mas dificil pero cuando me puse a investigar, lei que las "Listas"
que usaba en python ya era una pila e incluso tenia el comando pop que nunca habia utilizado, gracias a esto fue mas
sencillo de resolver la problematica
\newpage \section{Bibliografia}
Hopcroft, J. E. (1979). Introduction to Automata Theory, Languages, and Computation [Paperback] [Jan 01, 1979] John E.
Hopcroft (2.a ed., Vol. 2). Pearson.
Pila. (s. f.). [Ilustración]. programacionpython. https://programacionpython80889555.files.wordpress.com/2019/01/pilas.png

```