

# Maquina de Turing que duplica la cantidad de unos

Barrera Pérez Carlos Tonatihu  
Profesor: Genaro Juárez Martínez  
Computing Selected Topics  
Grupo: 3CM8

10 de septiembre de 2018

# Índice

1. Introducción	3
2. Desarrollo	4
3. Pruebas	7
Referencias	12

# 1. Introducción

La elaboración de este programa consistió en elaborar un maquina de Turing capaz de duplicar la cantidad de unos en una cadena de unos, es decir, si la cadena que se ingresa es 11 entonces la cadena de salida sera 1111.

Es por esto que la maquina sólo aceptara unos en la entrada mientras que los símbolos de la cinta incluirán a la X y la Y. De esta forma la maquina de Turing para este problema se define como:

$$M = (\{q_0, q_1, q_2, q_3\}, \{1\}, \{1, X, Y, B\}, \delta, q_0, B, \{q_4\})$$

donde  $\delta$  se especifica en la siguiente tabla:

	Símbolo			
Estado	1	X	Y	B
0	(1, X, R)	-	(3, 1, R)	-
1	(1, 1, R)	-	(1, Y, R)	(1, Y, L)
2	(2, 1, L)	(0, 1, R)	(2, Y, L)	-
3	-	-	(3, 1, R)	-

EL funcionamiento de esta maquina se puede entender mejor con el diagrama de la figura 1

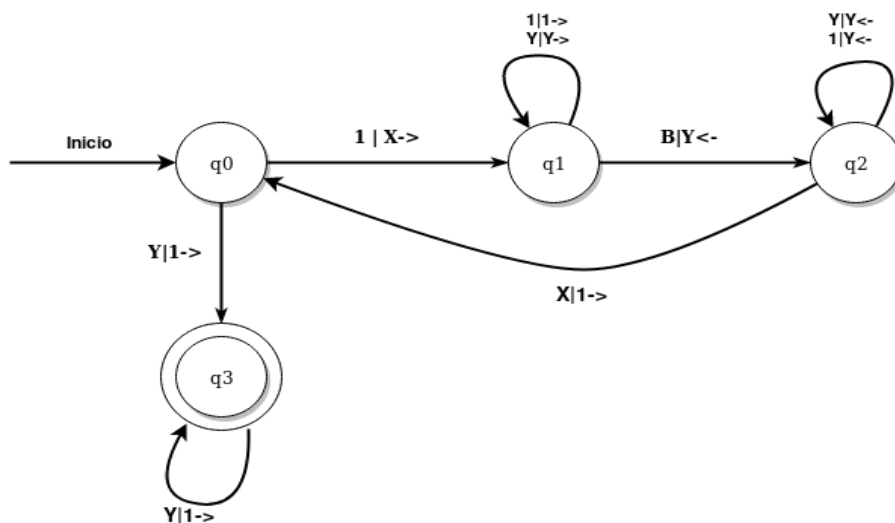


Figura 1: Representación gráfica de las transiciones de la maquina de Turing

El funcionamiento del programa empieza cuando se inserta una cadena de unos y la maquina empezara a trabajar, además de generar una cadena final se muestra una animación del como se están realizando las transiciones en la cinta de la maquina, por otro lado se imprime en consola el historial de movimientos que se hacen, este historial también se guarda en un archivo de texto.

## 2. Desarrollo

El código de este programa fue realizado en Python 3.7 y se utilizo la biblioteca tkinter para la parte gráfica.

Archivo: turing.py Esta clase es la que modela la maquina de Turing, los parámetros importantes son: el estado inicial, los estados finales, la cadena de entrada y la función de transición.

```
1 class MaquinaTuring:
2     """MaquinaTuring"""
3     def __init__(self, estado_inicial, estados_finales, cadena,
4         transiciones):
5         self.transiciones = transiciones
6         self.inicial = estado_inicial
7         self.finales = estados_finales
8         self.cinta = list(cadena)
9         self.estado_actual = self.inicial
10        self.apuntador = 0
11        self.blanco = "B"
12        self.direccion = None
13
14    def consumir(self):
15        """Toma un simbolo de la cinta y lo evalua en la funcion
16        de
17        transicion"""
18        if len(self.cinta) - 1 < self.apuntador:
19            caracter = 'B'
20        else:
21            caracter = self.cinta[self.apuntador]
22        tupla = (self.estado_actual, caracter)
23        if tupla in self.transiciones:
24            siguiente = self.transiciones[tupla]
25            if len(self.cinta) - 1 < self.apuntador:
26                self.cinta.append(self.blanco)
27            if self.apuntador < 0:
28                self.cinta.insert(0, self.blanco)
29            self.cinta[self.apuntador] = siguiente[1]
30            if siguiente[2] == "R":
31                self.apuntador += 1
```

```

30         else:
31             self.apuntador -= 1
32
33             self.estado_actual = siguiente[0]
34             self.direccion = siguiente[2]
35             return True
36     else:
37         return False
38
39     def es_final(self):
40         """Metodo para comprobar si nos encontramos en un estado
41         final"""
42         if self.estado_actual in self.finales:
43             if len(self.cinta) - 1 < self.apuntador or self.
44                 apuntador < 0:
45                 return True
46             return False

```

Archivo: diagrama.py Este archivo implementa la clase MaquinaTuring.py y se declaran los parámetros que se pasaran a este archivo así como la captura de la cadena y la escritura del registro de transiciones en consola y en archivo de texto, sin olvidar la animación de dichas transiciones.

```

1  import tkinter as tk
2  import time
3  from tkinter import font as tkfont
4  from turing import MaquinaTuring
5
6  # Tabla de transiciones que modela el automata
7  transiciones = {
8      ("q0", "1"): ("q1", "X", "R"),
9      ("q0", "Y"): ("q3", "1", "R"),
10     ("q1", "1"): ("q1", "1", "R"),
11     ("q1", "Y"): ("q1", "Y", "R"),
12     ("q1", "B"): ("q2", "Y", "L"),
13     ("q2", "1"): ("q2", "1", "L"),
14     ("q2", "X"): ("q0", "1", "R"),
15     ("q2", "Y"): ("q2", "Y", "L"),
16     ("q3", "Y"): ("q3", "1", "R"),
17 }
18
19 entrada = input("Ingrese la cadena de unos: ")
20 maquina = MaquinaTuring("q0", "q3", entrada, transiciones)
21
22 # Configuracion de la ventana
23 gui = tk.Tk()
24 gui.geometry("600x400+100+100")
25 gui.title("Maquina de Turing")
26 c = tk.Canvas(gui, width=600, height=400)

```

```

27 c.pack()
28 bold_font = tkfont.Font(family="Arial", size=24)
29
30 # Principales componentes que se animan
31 control = c.create_rectangle(150, 100, 200, 150, fill="lightblue
    ")
32 flecha = c.create_line(175, 150, 175, 175, arrow=tk.LAST, width
    =3)
33 texto = c.create_text(165, 200, text=''.join(maquina.cinta),
    font=bold_font,
34                        anchor=tk.W)
35 estado = c.create_text(160, 125, text=maquina.estado_actual,
    font=bold_font,
36                        anchor=tk.W)
37
38 archivo = open("salida.txt", "w+")
39 # Mientras no llegues a un estado final continua
40 while not maquina.es_final():
41     print('Cadena: {}'.format(''.join(maquina.cinta)))
42     print('Estado actual: {}, apuntador: {}'.format(maquina.
    estado_actual,
43             maquina.apuntador+1))
44
45     archivo.write('Cadena: {}\n'.format(''.join(maquina.cinta)))
46     archivo.write('Estado actual: {}, apuntador: {}\n'
47             .format(maquina.estado_actual, maquina.
    apuntador+1))
48     if not maquina.consumir():
49         print('*' * 20)
50         archivo.write('*' * 20)
51         archivo.write('\n')
52         break
53     print('Siguiente estado: {}'.format(maquina.estado_actual))
54     print('*'*20)
55
56     archivo.write('Siguiente estado: {}\n'.format(maquina.
    estado_actual))
57     archivo.write('*' * 20)
58     archivo.write('\n')
59
60     gui.update()
61     time.sleep(1)
62     c.itemconfigure(texto, text=''.join(maquina.cinta), anchor=
    tk.W)
63     c.itemconfigure(estado, text=maquina.estado_actual)
64     if maquina.direccion == 'R':
65         c.move(control, 19, 0)
66         c.move(flecha, 19, 0)
67         c.move(estado, 19, 0)

```

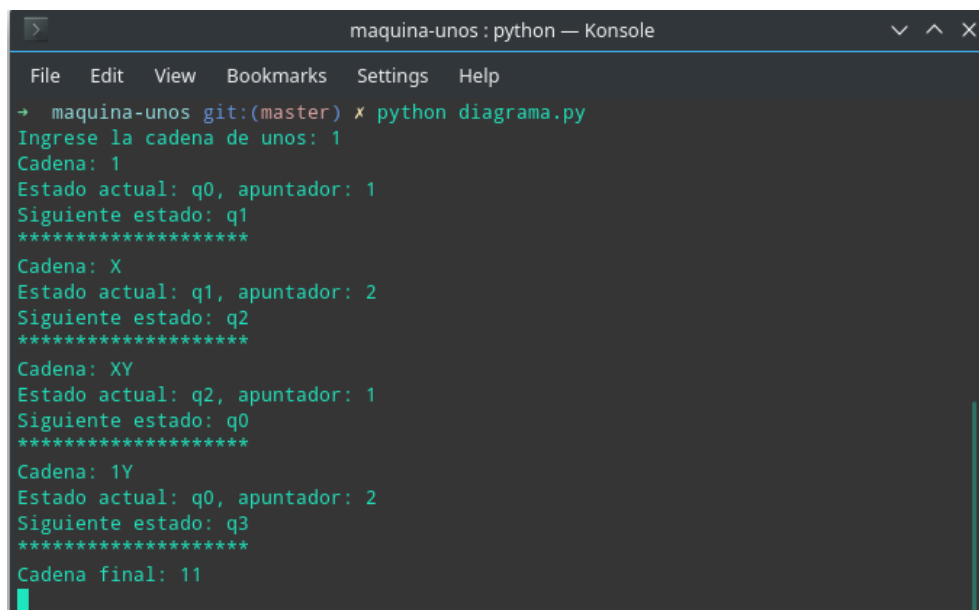
```

68     else:
69         c.move(control, -19, 0)
70         c.move(estado, -19, 0)
71         c.move(flecha, -19, 0)
72
73 print('Cadena final: {}'.format(''.join(maquina.cinta)))
74 archivo.write('Cadena final: {}\n'.format(''.join(maquina.cinta)
75 ))
76 archivo.close()
77 gui.mainloop()

```

### 3. Pruebas

El siguiente ejemplo es la cadena con un solo uno.



```

maquina-unos: python — Konsole
File Edit View Bookmarks Settings Help
→ maquina-unos git:(master) x python diagrama.py
Ingrese la cadena de unos: 1
Cadena: 1
Estado actual: q0, apuntador: 1
Siguiendo estado: q1
*****
Cadena: X
Estado actual: q1, apuntador: 2
Siguiendo estado: q2
*****
Cadena: XY
Estado actual: q2, apuntador: 1
Siguiendo estado: q0
*****
Cadena: 1Y
Estado actual: q0, apuntador: 2
Siguiendo estado: q3
*****
Cadena final: 11

```

Figura 2: Salida en consola

```
diagrama.py x salida.txt x turing.py x
1 Cadena: 1
2 Estado actual: q0, apuntador: 1
3 Siguiente estado: q1
4 *****
5 Cadena: X
6 Estado actual: q1, apuntador: 2
7 Siguiente estado: q2
8 *****
9 Cadena: XY
10 Estado actual: q2, apuntador: 1
11 Siguiente estado: q0
12 *****
13 Cadena: 1Y
14 Estado actual: q0, apuntador: 2
15 Siguiente estado: q3
16 *****
17 Cadena final: 11
18
```

Figura 3: Registro de transiciones

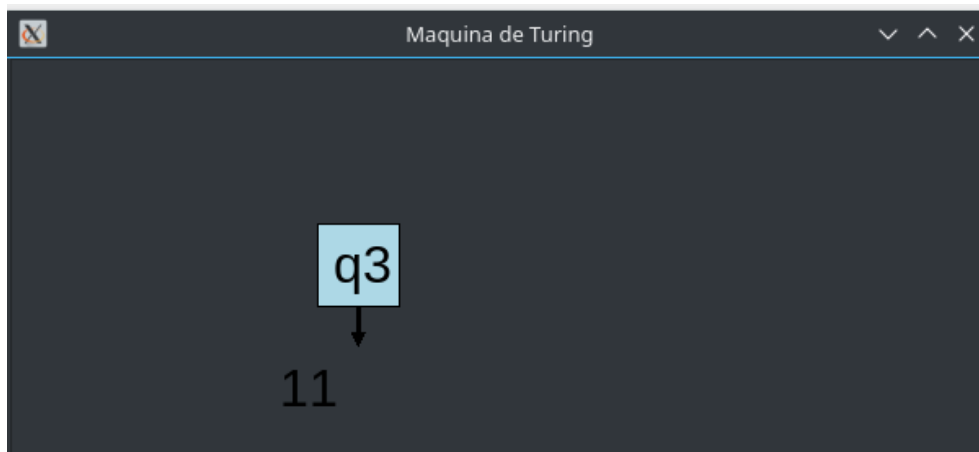


Figura 4: Representación gráfica de las transiciones de la maquina de Turing

En este ejemplo se inserta una cadena con dos unos.



```
maquina-unos : python — Konsole

File Edit View Bookmarks Settings Help

→ maquina-unos git:(master) python diagrama.py
Ingrese la cadena de unos: 11
Cadena: 11
Estado actual: q0, apuntador: 1
Siguiendo estado: q1
*****
Cadena: X1
Estado actual: q1, apuntador: 2
Siguiendo estado: q1
*****
Cadena: X1
Estado actual: q1, apuntador: 3
Siguiendo estado: q2
*****
Cadena: X1Y
Estado actual: q2, apuntador: 2
Siguiendo estado: q2
*****
Cadena: X1Y
Estado actual: q2, apuntador: 1
Siguiendo estado: q0
*****
Cadena: 11Y
Estado actual: q0, apuntador: 2
Siguiendo estado: q1
*****
Cadena: 1XY
Estado actual: q1, apuntador: 3
Siguiendo estado: q1
*****
Cadena: 1XY
Estado actual: q1, apuntador: 4
Siguiendo estado: q2
*****
Cadena: 1XY
Estado actual: q2, apuntador: 3
Siguiendo estado: q2
*****
Cadena: 1XY
Estado actual: q2, apuntador: 2
Siguiendo estado: q0
```

Figura 5: Salida en consola parte uno

```
*****
Cadena: 1XY
Estado actual: q2, apuntador: 3
Siguiete estado: q2
*****
Cadena: 1XY
Estado actual: q2, apuntador: 2
Siguiete estado: q0
*****
Cadena: 11Y
Estado actual: q0, apuntador: 3
Siguiete estado: q3
*****
Cadena: 111Y
Estado actual: q3, apuntador: 4
Siguiete estado: q3
*****
Cadena final: 1111
```

Figura 6: Salida en consola parte dos

```
diagrama.py x salida.txt x
1 Cadena: ·11
2 Estado actual: ·q0, ·apuntador: ·1
3 Siguiete estado: ·q1
4 *****
5 Cadena: ·X1
6 Estado actual: ·q1, ·apuntador: ·2
7 Siguiete estado: ·q1
8 *****
9 Cadena: ·X1
10 Estado actual: ·q1, ·apuntador: ·3
11 Siguiete estado: ·q2
12 *****
13 Cadena: ·X1Y
14 Estado actual: ·q2, ·apuntador: ·2
15 Siguiete estado: ·q2
16 *****
17 Cadena: ·X1Y
18 Estado actual: ·q2, ·apuntador: ·1
19 Siguiete estado: ·q0
20 *****
21 Cadena: ·11Y
22 Estado actual: ·q0, ·apuntador: ·2
23 Siguiete estado: ·q1
24 *****
25 Cadena: ·1XY
26 Estado actual: ·q1, ·apuntador: ·3
27 Siguiete estado: ·q1
28 *****
29 Cadena: ·1XY
30 Estado actual: ·q1, ·apuntador: ·4
31 Siguiete estado: ·q2
32 *****
33 Cadena: ·1XYY
34 Estado actual: ·q2, ·apuntador: ·3
35 Siguiete estado: ·q2
36 *****
37 Cadena: ·1XYY
38 Estado actual: ·q2, ·apuntador: ·2
39 Siguiete estado: ·q0
40 *****
41 Cadena: ·11YY
42 Estado actual: ·q0, ·apuntador: ·3
43 Siguiete estado: ·q3
44 *****
45 Cadena: ·111Y
46 Estado actual: ·q3, ·apuntador: ·4
47 Siguiete estado: ·q3
48 *****
49 Cadena final: ·1111
50
```

Figura 7: Registro de transiciones

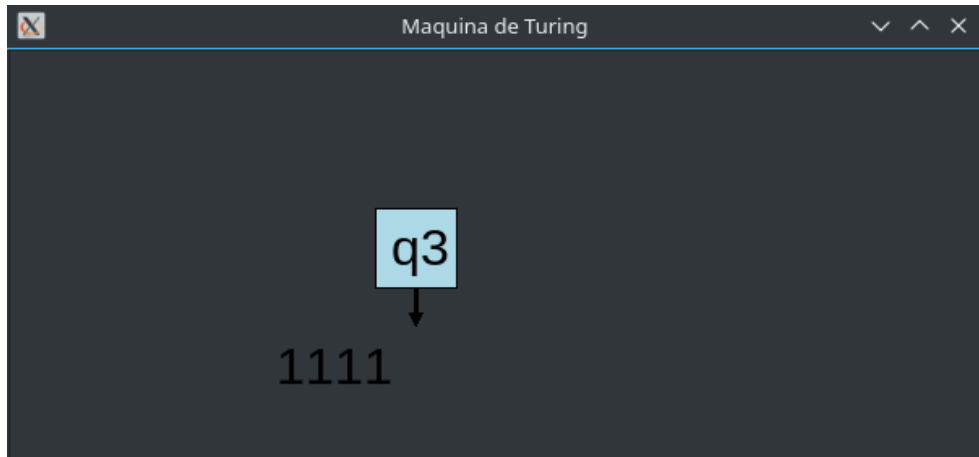


Figura 8: Representación gráfica de las transiciones de la maquina de Turing

## Referencias

- [1] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introducción a La Teoría De Autómatas, Lenguajes Y Computación*. Addison-Wesley, 2007.