



REPORTE

Programa 9 - máquina de Turing

Materia : Teoría de la Computación

Grupo : 4CM1

Alumno : Julio Cesar Hernández Reyes

Docente : Juárez Martínez Genaro

1. Introducción

En este reporte se explicara como se realizo el programa 9 - máquina de Turing, el cual es un programa que replica la maquina de Turing para la adición cortes, del paper: What can we learn from universal Turing machines? escrito por: Maurice Margenstern el 19 de Octubre del 2021.

El programa realiza lo que se especifica en la tabla de la página numero 2 del paper, que en si es para hacer la suma de n cantidad de l que se encuentran a la izquierda de un * con m cantidad de l que se encuentran a la derecha del mismo *. La estructura de las cadenas aceptadas por esta máquina de Turing es la siguiente:

$$_ * |^n * |^m * _$$

El funcionamiento del programa es que este evalúa una cadena de acuerdo a la tabla del paper, para que al final quede una nueva cadena formada por en primera la cadena original, y después de esta se concatena con la cadena formada por la suma de los l de cada lado de la cadena inicial, quedando de la siguiente manera:

$$_ * |^n * |^m * |^{n+m} * _$$

Los _ son solo para identificar espacios en blanco en la cadena, los * son separadores entre los l de la izquierda y la derecha. El Programa tiene un menú en el cual se puede ingresar la cadena a evaluar de forma manual, que la cadena se genere de manera automática, y salir del programa.

Las funcionalidades extra es que la historia del cada iteracion entre la cadena a evaluar y los estados de la tabla se guarda en un archivo .txt el cual al final igual de cada opción si la cadena a evaluar es de menos de 10 caracteres, la historia se imprime por consola pero con una pequeña animación del tipo maquina de escribir y además en color verde, para ir viendo que se hizo en la evaluación de la cadena.

Para la realización del programa se uso Python. Se uso un IDE en vez de compilar y ejecutar por consola, esto para una mayor facilidad a la hora de corregir errores y algunos detalles del programa.

2. Información Importante

2.1. Tabla de la máquina de Turing

Esta es la tabla del paper que se simulo con el programa:

	—	*		a	X
1		X R 2			
2		R 3	R		
3		X L 4	R		
4		L	a R 5		R 7
5	L 6	R	R		R
6		L	L	L 4	L
7		R 8	R		
8	* L 9		R		* R
9		L	L		* !

La notación usada en la tabla sigue las convenciones de Minsky. El formato para la instrucción es el de AMs en donde la A es la letra escrita por la máquina en el espacio escaneado en el lugar de la letra que estaba antes, la M es el movimiento del cabezal, ya sea a la izquierda, designado por L, o a la derecha, designado por R. El símbolo ! significa el estado de paro, cuando se para la computación, o el análisis de la cadena en pocas palabras.

3. Desarrollo

3.1. Código del programa

Código creado con Python en el IDE de Pycharm:

```
1 # PROGRAMA 9
2 # Del art culo de arXiv "What can we learn from universal Turing
  machines?"
3 # Se tiene que programar la m quina de la tabla 1
4
5 # 1.La m quina se tiene que animar para cadenas peque as (<10
  caracteres) LISTO
6 # 2.Puede recibir la cadena por parte del usuario o aleatoriamente
  LISTO
7 # 3.Mandar la salida a un archivo de texto que muestre las
  descripciones LISTO
```

```

8 # instantneas por rengl n en cada iteraci n
9 import os
10 import sys
11 import time
12 from termcolor import colored
13 import random
14
15
16 def limpiarpantalla():
17     if os.name == "posix":
18         os.system("clear")
19     elif os.name == "ce" or os.name == "nt" or os.name == "dos":
20         os.system("cls")
21
22 def crearcadena(izquierda, derecha):
23     cad_izquierda = ""
24     cad_derecha = ""
25     conta = 0
26     while conta < izquierda:
27         cad_izquierda = cad_izquierda + '|'
28         conta += 1
29     conta = 0
30     while conta < derecha:
31         cad_derecha = cad_derecha + '|'
32         conta += 1
33     cadena_final = '_*' + cad_izquierda + '*' + cad_derecha + '*_'
34     return cadena_final
35
36
37 def sustituirletra(conta, cadena, letra):
38     lista = list(cadena)
39     lista[conta] = letra
40     cadena = ''.join(lista)
41     return cadena
42
43
44 def TuringMachine(cadena):
45     q1 = 1
46     q2 = 2
47     q3 = 3
48     q4 = 4
49     q5 = 5
50     q6 = 6
51     q7 = 7
52     q8 = 8

```

```

53 q9 = 9
54 conta = 1
55 estado_atual = q1
56 #print(cadena, conta, estado_atual)
57 historia(cadena, conta, estado_atual)
58 while 1:
59     if estado_atual == q1:
60         #print(cadena, conta, estado_atual)
61         if cadena[conta] == '*':
62             cadena = sustituirletra(conta, cadena, 'X')
63             conta = conta + 1
64             estado_atual = q2
65             historia(cadena, conta, estado_atual)
66     if estado_atual == q2:
67         #print(cadena, conta, estado_atual)
68         if cadena[conta] == '*':
69             conta += 1
70             estado_atual = q3
71             historia(cadena, conta, estado_atual)
72         elif cadena[conta] == '|':
73             conta += 1
74             historia(cadena, conta, estado_atual)
75     if estado_atual == q3:
76         #print(cadena, conta, estado_atual)
77         if cadena[conta] == '*':
78             cadena = sustituirletra(conta, cadena, 'X')
79             conta -= 1
80             estado_atual = q4
81             historia(cadena, conta, estado_atual)
82         elif cadena[conta] == '|':
83             conta = conta + 1
84             historia(cadena, conta, estado_atual)
85     if estado_atual == q4:
86         #print(cadena, conta, estado_atual)
87         if cadena[conta] == '*':
88             conta = conta - 1
89             historia(cadena, conta, estado_atual)
90         elif cadena[conta] == '|':
91             cadena = sustituirletra(conta, cadena, 'a')
92             conta = conta + 1
93             estado_atual = q5
94             historia(cadena, conta, estado_atual)
95         elif cadena[conta] == 'X':
96             conta = conta + 1
97             estado_atual = q7

```

```

98         historia(cadena, conta, estado_actual)
99     if estado_actual == q5:
100         #print(cadena, conta, estado_actual)
101         if cadena[conta] == '_':
102             cadena = sustituirletra(conta, cadena, '|')
103             cadena = cadena + '_'
104             conta = conta - 1
105             estado_actual = q6
106             historia(cadena, conta, estado_actual)
107         elif cadena[conta] == '*':
108             conta = conta + 1
109             historia(cadena, conta, estado_actual)
110         elif cadena[conta] == '|':
111             conta = conta + 1
112             historia(cadena, conta, estado_actual)
113         elif cadena[conta] == 'X':
114             conta = conta + 1
115             historia(cadena, conta, estado_actual)
116     if estado_actual == q6:
117         #print(cadena, conta, estado_actual)
118         if cadena[conta] == '*':
119             conta = conta - 1
120             historia(cadena, conta, estado_actual)
121         elif cadena[conta] == '|':
122             conta = conta - 1
123             historia(cadena, conta, estado_actual)
124         elif cadena[conta] == 'a':
125             cadena = sustituirletra(conta, cadena, '|')
126             conta = conta - 1
127             estado_actual = q4
128             historia(cadena, conta, estado_actual)
129         elif cadena[conta] == 'X':
130             conta = conta - 1
131             historia(cadena, conta, estado_actual)
132     if estado_actual == q7:
133         #print(cadena, conta, estado_actual)
134         if cadena[conta] == '*':
135             conta = conta + 1
136             estado_actual = q8
137             historia(cadena, conta, estado_actual)
138         elif cadena[conta] == '|':
139             conta = conta + 1
140             historia(cadena, conta, estado_actual)
141     if estado_actual == q8:
142         #print(cadena, conta, estado_actual)

```

```

143         if cadena[conta] == '_':
144             cadena = sustituirletra(conta, cadena, '*')
145             cadena = cadena + '_'
146             conta = conta - 1
147             estado_actual = q9
148             historia(cadena, conta, estado_actual)
149         elif cadena[conta] == '|':
150             conta = conta + 1
151             historia(cadena, conta, estado_actual)
152         elif cadena[conta] == 'X':
153             cadena = sustituirletra(conta, cadena, '*')
154             conta = conta + 1
155             historia(cadena, conta, estado_actual)
156     if estado_actual == q9:
157         #print(cadena, conta, estado_actual)
158         if cadena[conta] == '*':
159             conta = conta - 1
160             historia(cadena, conta, estado_actual)
161         elif cadena[conta] == '|':
162             conta = conta - 1
163             historia(cadena, conta, estado_actual)
164         elif cadena[conta] == 'X':
165             cadena = sustituirletra(conta, cadena, '*')
166             historia(cadena, conta, estado_actual)
167         #return cadena
168         break
169
170 def manual():
171     limpiarpantalla()
172     print('Manual')
173     try:
174         izquierda = input('Cantidad de | del lado izquierdo:')
175         derecha = input('Cantidad de | del lado derecho:')
176         cadena_creada = crearcadena(int(izquierda), int(derecha))
177         suma = int(izquierda) + int(derecha)
178     except ValueError:
179         izquier = 0
180         dere = 0
181         cadena_creada = crearcadena(izquier, dere)
182         suma = 0
183     print('Cadena creada: ', end='')
184     print(cadena_creada)
185     print(" ")
186     tituloarchivo()
187     TuringMachine(cadena_creada)

```

```

188     print('Cadena evaluada, Archivo de historia creado')
189     if suma <= 10:
190         animacion()
191     input()
192
193
194 def automatico():
195     limpiarpantalla()
196     print('Automático')
197     izquierda = random.randint(0,100)
198     derecha = random.randint(0,100)
199     suma = izquierda + derecha
200     cadena_creada = crearcadena(izquierda, derecha)
201     print('Cadena creada: ', end='')
202     print(cadena_creada)
203     print(" ")
204     tituloarchivo()
205     TuringMachine(cadena_creada)
206     print('Cadena evaluada, Archivo de historia creado')
207     if suma <= 10:
208         animacion()
209     input()
210
211 def tituloarchivo():
212     archivo = open("Historia.txt", "a")
213     linea = "Cadena      Estado      Conta"
214     archivo.write(linea)
215     archivo.write("\n")
216     archivo.close()
217
218
219 def historia(cadena, conta, estado):
220     archivo = open("Historia.txt", "a")
221     linea = cadena + " " + str(estado) + " " + str(conta)
222     archivo.write(linea)
223     archivo.write("\n")
224     archivo.close()
225
226
227 def animacion():
228     archivo = open("Historia.txt", "r")
229     while True:
230         linea = archivo.readline()
231         for char in linea:
232             sys.stdout.write(colored(char, 'green'))

```

```

233         sys.stdout.flush()
234         if char != "\n":
235             time.sleep(0.1)
236     if not linea:
237         break
238     archivo.close()
239
240
241 def menu():
242     if os.path.isfile("Historia.txt"):
243         os.remove("Historia.txt")
244     while True:
245         limpiarpantalla()
246         print('Menu Programa 9')
247         print('1.Forma Manual')
248         print('2.Forma Automatica')
249         print('3.Salir')
250         respuesta = input('Favor de escoger una opci n \n')
251         if respuesta == '1':
252             if os.path.isfile("Historia.txt"):
253                 os.remove("Historia.txt")
254             manual()
255         elif respuesta == '2':
256             if os.path.isfile("Historia.txt"):
257                 os.remove("Historia.txt")
258             automatico()
259         elif respuesta == '3':
260             break
261         else:
262             print('Igrese un valor entre 1 y 3')
263             input()
264     print('Fin del Programa')
265
266
267 if __name__ == '__main__':
268     menu()

```

3.2. Explicación

El programa tiene la función principal de evaluar una cadena de caracteres de acuerdo a la tabla antes revisada, las funciones usadas tienen las siguientes funcionalidades:

`limpiarpantalla()`:

Es la función que limpia la consola/pantalla cuando se necesite.

`crearcadena(izquierda, derecha):`

Esta función crea una cadena con el formato usado para la evaluación con lo que dice la tabla de la máquina de Turing. Solo necesita la cantidad de `|` en cada lado de la cadena para que se forme la cadena solicitada.

`sustituirletra(conta, cadena, letra):`

Esta función sustituye una letra/símbolo en la posición especificada de la cadena ingresada.

`TuringMachine(cadena):`

Es la encargada de hacer la evaluación de la cadena de acuerdo a la tabla del paper, así como manejar la historia de cada acción que se tome dependiendo la cadena, en el archivo `Historia.txt`, simula los 9 estados de la tabla y sus respectivas acciones.

`manual():`

Esta función hace el pedido de las cantidades de `|` dependiendo de cada lado. Si no se ingresa alguna cantidad el default es de 0 en cada lado. de esta forma se puede crear la cadena con los datos ingresados, y luego meterla a evaluar en la máquina de Turing, también llama a `animacion()`

`automatico():`

Esta función hace exactamente lo mismo que la de `manual()` pero en esta la cadena se crea con cantidades aleatorias de `|` en cada lado, el rango decidido es de 0 a 100 caracteres por lado.

`tituloarchivo():`

Le pone al archivo de `Historia.txt` que valores son los de cada columna: Cadena Estado Conta historia(estsiguiente, estactual, letra):

En esta función se guarda la historia del automata en un `archivo.txt` de cada cadena que se haya evaluado en el autómata.

`historia():`

Es la función que va guardando cada acción de la máquina de Turing en el archivo `Historia.txt`

`animacion():`

Es la función que crea la animación por consola de la historia de la cadena ingresada solo si esta tiene menos de 10 caracteres (`|`) en total.

`menu():`

Es la función que despliega el menú del programa con las opciones de forma manual, forma automática, y salir del programa.

3.3. Ejecución del programa, en Python

Ejecución del programa por consola de Pycharm:

```
Terminal: Local x + v
cesar@cesar-HP-Notebook:~/Documentos/Teoria de la Computación/Programa_9/python$ python3 main.py
```

Menú del programa:

```
Terminal: Local x + v
Menu Programa 9
1.Forma Manual
2.Forma Automatica
3.Salir
Favor de escoger una opción
```

Opción 1. Forma Manual:

```
Terminal: Local x + v
Menu Programa 9
1.Forma Manual
2.Forma Automatica
3.Salir
Favor de escoger una opción
1
```

```

Terminal: Local x + v
Manual
Cantidad de | del lado izquierdo:2
Cantidad de | del lado derecho:3
Cadena creada: _*||*||*_

Cadena evaluada, Archivo de historia creado
Cadena      Estado      Conta
_*||*||

```

Animación tipo maquina de escribir pues la cadena fue de menos de 10 caracteres:

```

Terminal: Local x + v
Cadena      Estado      Conta
_*||*||*_   1  1
_X||*||*_   2  2
_X||*||*_   2  3
_X||*||*_   2  4
_X||*||*_   3  5
_X||*||*_   3  6
_X||*||*_   3  7
_X||*||*_   3  8
_X||

```

```

Terminal: Local x + v
_X||*||*||*||*||*_   9  8
_X||*||*||*||*||*_   9  7
_X||*||*||*||*||*_   9  6
_X||*||*||*||*||*_   9  5
_X||*||*||*||*||*_   9  4
_X||*||*||*||*||*_   9  3
_X||*||*||*||*||*_   9  2
_X||*||*||*||*||*_   9  1
_*||*||*||*||*||*_   9  1

```

Archivo de Historia creado para la cadena creada:

```

main.py x Historia.txt x
99      _X||*||*||*||*||*_   9  8
100     _X||*||*||*||*||*_   9  7
101     _X||*||*||*||*||*_   9  6
102     _X||*||*||*||*||*_   9  5
103     _X||*||*||*||*||*_   9  4
104     _X||*||*||*||*||*_   9  3
105     _X||*||*||*||*||*_   9  2
106     _X||*||*||*||*||*_   9  1
107     _*||*||*||*||*||*_   9  1
108

```

Opción 2. Forma Automática:

```
Terminal: Local x + v  
Menu Programa 9  
1.Forma Manual  
2.Forma Automatica  
3.Salir  
Favor de escoger una opción  
2
```

```
Terminal: Local x + v  
Automático  
Cadena creada: _*|*****|*|*****|*_  
  
Cadena evaluada, Archivo de historia creado
```

Archivo de Historia creado para la cadena creada:

[illegible]

Opción 3. Salir del programa:

```
Terminal: Local x + v
Menu Programa 9
1.Forma Manual
2.Forma Automatica
3.Salir
Favor de escoger una opción
3
Fin del Programa
cesar@cesar-HP-Notebook:~/Documentos/Teoria de la Computación/Programa_9/python$
```

4. Conclusiones

Este programa no lo entendí en un principio por la notación que se manejaba en el paper, pero al leerlo varias veces y entender como funcionaba la tabla, hice algunas pruebas en papel para ver como funcionaba, lo que al final resulto, pues se me hizo muy fácil comprender el funcionamiento y poder empezar a programar lo solicitado, la maquina de turing simulada es muy interesante y es la base para otras maquinas, aunque la de este programa solo manejaba el símbolo $|$ puede ser cualquiera.