

Máquina de Turing

Sebastián Acevedo Juárez
Teoría de la computación

June 2022

1 Introducción

En este reporte se muestra la resolución del problema de la Máquina de Turing, en donde tendremos como entrada una cadena de tipo $*|{}^n * |{}^m *$ y obtendremos a la salida una cadena con la estructura $*|{}^n * |{}^m * |{}^{n+m}$.

2 Marco Teórico

Una maquina de Turing consiste de un control finito que puede estar en cualquier estado de un conjunto finito de estados. Se tiene una cinta dividida en celdas, cada celda con un símbolo. Inicialmente, la entrada (cadena finita de símbolos del alfabeto) se coloca en la cinta, el resto de las celdas tienen el símbolo especial vacío. La cabeza de la cinta esta siempre sobre una celda y al principio esta sobre la celda más a la izquierda con el primer símbolo de la cadena de entrada. Un movimiento o transición puede cambiar de estado (o quedarse en el estado actual), escribir un símbolo(reemplazando el símbolo que existía o dejando el mismo) y mover la cabeza a la izquierda o derecha. Una Máquina de Turing consta de:

1. Q : es un conjunto finito de estados
2. Σ : es un conjunto finito de símbolos de entrada
3. Γ : es el conjunto de símbolos de la cinta ($\Sigma \cup \Gamma$)
4. δ : la función de transición $\delta(q, X) = (p, Y, D)$, donde p es el siguiente estado en Q , Y es el símbolo en Γ que se escribe en la celda que esta viendo la cabeza de la cinta y D es la dirección (izq. o der.).
5. q_0 : es el estado inicial
6. B : es el símbolo de vacío, que esta en Γ pero no en Σ
7. F : es el conjunto de estados finales o de aceptación.

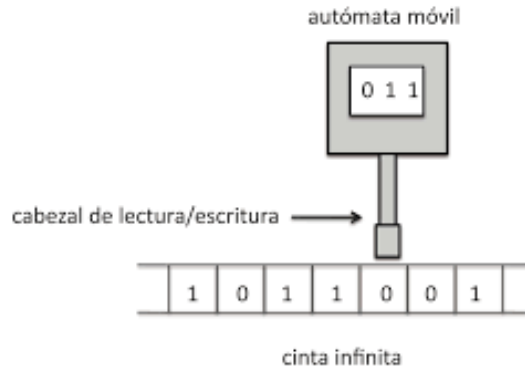


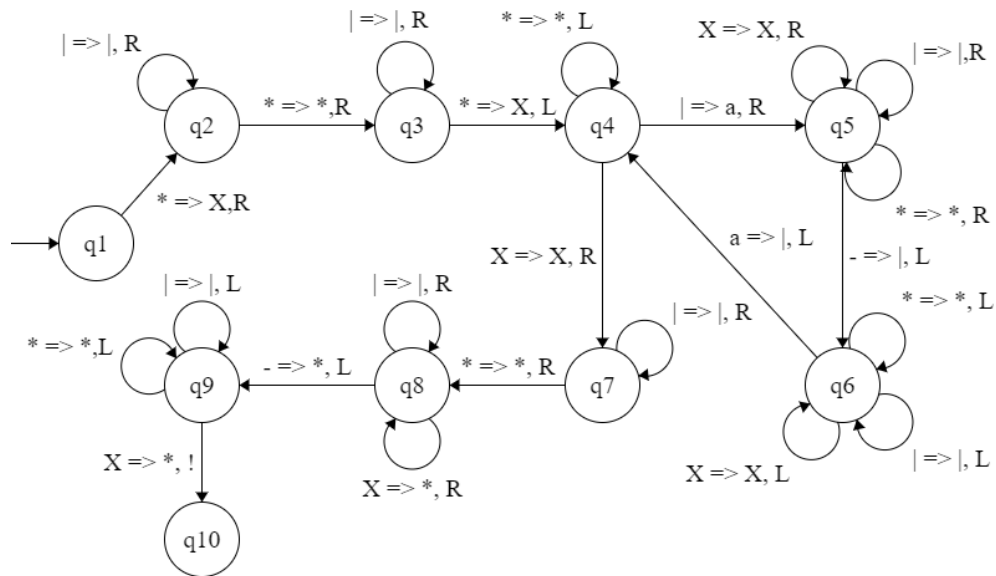
Figure 1: Representación gráfica de una Máquina de Turing

3 Desarrollo

En la implementación, hicimos uso de la tabla proporcionada en el problema para obtener el resultado obtenido. El programa básicamente ejecuta lo representado por la tabla, o en su defecto, por el grafo. en este último es más entendible y visual el camino que recorre la cadena dentro del programa. Al final, obtendremos una cadena de la forma $*|{}^n * |{}^m * |{}^{n+m}$.

Table 1 *Table of the Turing machine for the courteous addition.*

	-	*		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		* !	



Representación gráfica de la Máquina de Turing

```

1 from asyncio.windows_events import NULL
2 from ntpath import join
3 from pickle import TRUE
4 from random import randint
5
6
7 def main():
8     print("==== M QUINA DE TURING ===")
9     print("Seleccione un modo")
10    print("1.- Autom tico\n2.- Manual")
11    if int(input()) == 1:
12        n = randint(0,20)
13        m = randint(0,20)
14        cadena = "*" + n*"|" + "*" + m*"|" + "*"
15    else:
16        print("Ingrese la cadena")
17        cadena = input()
18        while(1):
19            if len(cadena) > 50:
20                print("Cadena inv lida")
21            else:
22                break
23    print("Cadena Original: ",cadena)
24    automata(list(cadena))
25
26

```

```

27 def automata(cadena):
28     tape = open("tape.txt", "w")
29     Right = 1
30     Left = -1
31     IDiS = open("IDS.txt", "w")
32
33     estado = 1
34     i = 0
35     while(1):
36         j = i
37         preves = estado
38         if i < len(cadena): prevchar = cadena[i]
39         else: prevchar = "-"
40         tape.write((i)*" "+v\n")
41         tape.write(''.join(cadena)+"\t\tq="+str(estado)+"\n\n\n")
42         if estado == 1:
43             if cadena[i] == "*":
44                 cadena[i] = "X"
45                 i += Right
46                 estado = 2
47             else:
48                 print("Cadena Inv lida")
49                 exit(0)
50
51         elif estado == 2:
52             if cadena[i] == "*":
53                 cadena[i] = "*"
54                 i += Right
55                 estado = 3
56             elif cadena[i] == "|":
57                 cadena[i] = "|"
58                 i += Right
59                 estado = 2
60             else:
61                 print("Cadena Inv lida")
62                 exit(0)
63
64         elif estado == 3:
65             if cadena[i] == "*":
66                 cadena[i] = "X"
67                 i += Left
68                 estado = 4
69             elif cadena[i] == "|":
70                 cadena[i] = "|"
71                 i += Right
72                 estado = 3
73             else:
74                 print("Cadena Inv lida")
75                 exit(0)
76
77         elif estado == 4:
78             if cadena[i] == "*":
79                 cadena[i] = "*"
80                 i += Left
81                 estado = 4
82             elif cadena[i] == "|":
83                 cadena[i] = "a"

```

```

84         i += Right
85         estado = 5
86     elif cadena[i] == "X":
87         cadena[i] = "X"
88         i += Right
89         estado = 7
90     else:
91         print("Cadena Inv lida")
92         exit(0)
93
94     elif estado == 5:
95         if i==len(cadena):
96             cadena.append("|")
97             i += Left
98             estado = 6
99         elif cadena[i] == "*":
100             cadena[i] = "*"
101             i += Right
102             estado = 5
103         elif cadena[i] == "|":
104             cadena[i] = "|"
105             i += Right
106             estado = 5
107         elif cadena[i] == "X":
108             cadena[i] = "X"
109             i += Right
110             estado = 5
111         else:
112             print("Cadena Inv lida")
113             exit(0)
114
115     elif estado == 6:
116         if cadena[i] == "*":
117             cadena[i] = "*"
118             i += Left
119             estado = 6
120         elif cadena[i] == "|":
121             cadena[i] = "|"
122             i += Left
123             estado = 6
124         elif cadena[i] == "a":
125             cadena[i] = "|"
126             i += Left
127             estado = 4
128         elif cadena[i] == "X":
129             cadena[i] = "X"
130             i += Left
131             estado = 6
132         else:
133             print("Cadena Inv lida")
134             exit(0)
135
136     elif estado == 7:
137         if cadena[i] == "*":
138             cadena[i] = "*"
139             i += Right
140             estado = 8

```

```

141     elif cadena[i] == "|":
142         cadena[i] = "|"
143         i += Right
144         estado = 7
145     else:
146         print("Cadena Inv lida")
147         exit(0)
148
149     elif estado == 8:
150         if i == len(cadena):
151             cadena.append("*")
152             i += Left
153             estado = 9
154         elif cadena[i] == "|":
155             cadena[i] = "|"
156             i += Right
157             estado = 8
158         elif cadena[i] == "X":
159             cadena[i] = "*"
160             i += Right
161             estado = 8
162         else:
163             print("Cadena Inv lida")
164             exit(0)
165
166     elif estado == 9:
167         if cadena[i] == "*":
168             cadena[i] = "*"
169             i += Left
170             estado = 9
171         elif cadena[i] == "|":
172             cadena[i] = "|"
173             i += Left
174             estado = 9
175         elif cadena[i] == "X":
176             cadena[i] = "*"
177             i = i
178             estado = 10
179         else:
180             print("Cadena Inv lida")
181             exit(0)
182     elif estado == 10:
183         print("Ejecuci n de la cadena terminada.\nCadena Final:",
184               end="")
185         print(''.join(cadena))
186         exit(0)
187
188     if j < i: #R
189         IDiS.write("q"+str(preves)+", "+prevchar+", "+cadena[j]+", R,
190                  q"+str(estado)+"\n")
191     elif j>i: #L
192         IDiS.write("q"+str(preves)+", "+prevchar+", "+cadena[j]+", L,
193                  q"+str(estado)+"\n")
194 tape.close()
195 IDiS.close()

```

193 main()

Algoritmo 1: Implementación de DFA

Ejecución Manual

```
==== MÁQUINA DE TURING ===  
Seleccione un modo  
1.- Automático  
2.- Manual  
2  
Ingresa la cadena  
*||*|*  
Cadena Original:  *||*|*  
Ejecución de la cadena terminada.  
Cadena Final:*||*|*|||*  
PS D:\Tablero\Programas>
```

```
1 q1, *, X, R, q2  
2 q2, |, |, R, q2  
3 q2, |, |, R, q2  
4 q2, *, *, R, q3  
5 q3, |, |, R, q3  
6 q3, *, X, L, q4  
7 q4, |, a, R, q5  
8 q5, X, X, R, q5  
9 q5, -, |, L, q6  
10 q6, X, X, L, q6  
11 q6, a, |, L, q4  
12 q4, *, *, L, q4  
13 q4, |, a, R, q5  
14 q5, *, *, R, q5  
15 q5, |, |, R, q5  
16 q5, X, X, R, q5
```

```

17 q5, l, l, R, q5
18 q5, -, l, L, q6
19 q6, l, l, L, q6
20 q6, X, X, L, q6
21 q6, l, l, L, q6
22 q6, *, *, L, q6
23 q6, a, l, L, q4
24 q4, l, a, R, q5
25 q5, l, l, R, q5
26 q5, *, *, R, q5
27 q5, l, l, R, q5
28 q5, X, X, R, q5
29 q5, l, l, R, q5
30 q5, l, l, R, q5
31 q5, -, l, L, q6
32 q6, l, l, L, q6
33 q6, l, l, L, q6
34 q6, X, X, L, q6
35 q6, l, l, L, q6
36 q6, *, *, L, q6
37 q6, l, l, L, q6
38 q6, a, l, L, q4
39 q4, X, X, R, q7
40 q7, l, l, R, q7
41 q7, l, l, R, q7
42 q7, *, *, R, q8
43 q8, l, l, R, q8
44 q8, X, *, R, q8
45 q8, l, l, R, q8
46 q8, l, l, R, q8
47 q8, l, l, R, q8
48 q8, -, *, L, q9
49 q9, l, l, L, q9
50 q9, l, l, L, q9
51 q9, l, l, L, q9
52 q9, *, *, L, q9
53 q9, l, l, L, q9
54 q9, *, *, L, q9
55 q9, l, l, L, q9
56 q9, l, l, L, q9

```

Algoritmo 2: Historial de estados

```

1 v
2 *|*|*|* q=1
3
4
5 v
6 X|*|*|* q=2
7
8
9 v
10 X|*|*|* q=2
11
12
13 v
14 X|*|*|* q=2
15

```



```

16
17      v
18 X || * | *      q=3
19
20
21      v
22 X || * | *      q=3
23
24
25      v
26 X || * | X      q=4
27
28
29      v
30 X || * a X      q=5
31
32
33      v
34 X || * a X      q=5
35
36
37      v
38 X || * a X |    q=6
39
40
41      v
42 X || * a X |    q=6
43
44
45      v
46 X || * | X |    q=4
47
48
49      v
50 X || * | X |    q=4
51
52
53      v
54 X | a * | X |    q=5
55
56
57      v
58 X | a * | X |    q=5
59
60
61      v
62 X | a * | X |    q=5
63
64
65      v
66 X | a * | X |    q=5
67
68
69      v
70 X | a * | X |    q=5
71
72

```

```

73      v
74 X|a*|X||    q=6
75
76
77      v
78 X|a*|X||    q=6
79
80
81      v
82 X|a*|X||    q=6
83
84
85      v
86 X|a*|X||    q=6
87
88
89      v
90 X|a*|X||    q=6
91
92
93      v
94 X||*|X||    q=4
95
96
97      v
98 Xa|*|X||    q=5
99
100
101      v
102 Xa|*|X||    q=5
103
104
105      v
106 Xa|*|X||    q=5
107
108
109      v
110 Xa|*|X||    q=5
111
112
113      v
114 Xa|*|X||    q=5
115
116
117      v
118 Xa|*|X||    q=5
119
120
121      v
122 Xa|*|X||    q=5
123
124
125      v
126 Xa|*|X||    q=6
127
128
129      v

```

```

130 Xa|*|X|||    q=6
131
132
133      v
134 Xa|*|X|||    q=6
135
136
137      v
138 Xa|*|X|||    q=6
139
140
141      v
142 Xa|*|X|||    q=6
143
144
145      v
146 Xa|*|X|||    q=6
147
148
149      v
150 Xa|*|X|||    q=6
151
152
153      v
154 X||*|X|||    q=4
155
156
157      v
158 X||*|X|||    q=7
159
160
161      v
162 X||*|X|||    q=7
163
164
165      v
166 X||*|X|||    q=7
167
168
169      v
170 X||*|X|||    q=8
171
172
173      v
174 X||*|X|||    q=8
175
176
177      v
178 X||*|*|||    q=8
179
180
181      v
182 X||*|*|||    q=8
183
184
185      v
186 X||*|*|||    q=8

```

```

187
188
189           v
190 X | | * | * | | |    q=8
191
192
193           v
194 X | | * | * | | | *    q=9
195
196
197           v
198 X | | * | * | | | *    q=9
199
200
201           v
202 X | | * | * | | | *    q=9
203
204
205           v
206 X | | * | * | | | *    q=9
207
208
209           v
210 X | | * | * | | | *    q=9
211
212
213           v
214 X | | * | * | | | *    q=9
215
216
217           v
218 X | | * | * | | | *    q=9
219
220
221           v
222 X | | * | * | | | *    q=9
223
224
225           v
226 X | | * | * | | | *    q=9
227
228
229           v
230 * | | * | * | | | *    q=10

```

Algoritmo 3: Muestra de la cinta

4 Conclusión

Gracias a este programa, pude terminar de comprender el tema de Máquina de Turing, ya que a un principio me costaba entender el tema, sin embargo, aplicando los conocimientos es más visual y didáctico para poder comprenderlo. Fue un problema más sencillo de lo que suena. Muchas veces las formalidades hacen ver más complicados a los problemas, cuando en realidad no lo son. Este problema representa el cómputo en su totalidad. Es impresionante pensar la cantidad de procesos similares que puede realizar una computadora actual y compararlo con los principios de la computación.

References

- [1] NAOE. (2009). Máquinas de Turing. Automatas. Recuperado 19 de junio de 2022, de <https://ccc.inaoep.mx/emorales/Cursos/Automatas/Intro-MaquinasTuring.pdf>