

Buscador de Palabras

Sebastián Acevedo Juárez
Teoría de la computación

June 2022

1 Introducción

El problema se nos plantea de la siguiente manera: Plantear un autómata que nos permita comprobar si las palabras "web, webpage, website, webmaster, ebay, page, site" se encuentran dentro de la cadena de entrada. Para esto, primero tendremos que plantear un NFA, para así convertirlo en un DFA que será el autómata que tendremos que programar.

2 Marco Teórico

Un **autómata finito determinista**, es la implementación de un autómata que sólo puede estar en un único estado después de leer cualquier secuencia de entradas. El término "determinista" hace referencia al hecho de que para cada entrada sólo existe uno y sólo un estado al que el autómata puede hacer la transición a partir de su estado actual. En palabras más simples, en un DFA (o AFD, por sus siglas en inglés y español respectivamente), sólo es posible seguir un camino por condición. Un Autómata Finito Determinista consta de:

1. Un conjunto finito de estados, a menudo designado como Q .
2. Un conjunto finito de símbolos de entrada, a menudo designado como Σ (sigma).
3. Una función de transición que toma como argumentos un estado y un símbolo de entrada y devuelve un estado. La función de transición se designa habitualmente como δ o Δ (delta).
4. Un estado inicial, uno de los estados de Q .
5. Un conjunto de estados finales o de aceptación F . El conjunto F es un subconjunto de Q .

La representación más sucinta de un AFD consiste en un listado de los cinco componentes anteriores. Normalmente, en las demostraciones, definiremos un AFD utilizando la notación de quintupla siguiente: $A = (Q, \Sigma, \delta, q_0, F)$

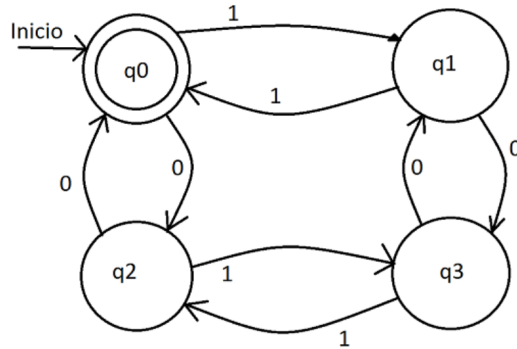
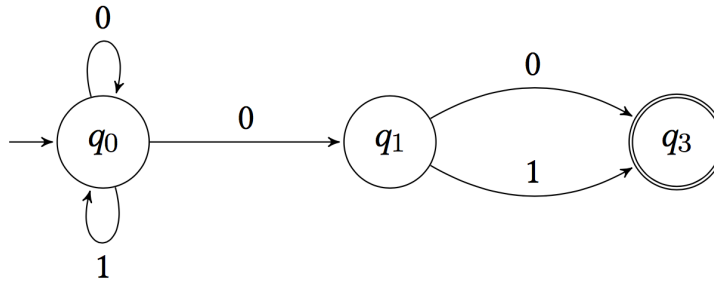


Figure 1: Representación gráfica de un DFA

Un **autómata finito no determinista** (NFA o AFN por sus siglas en inglés y español respectivamente), es el autómata finito que tiene transiciones vacías o que por cada símbolo desde un estado de origen se llega a más de un estado destino, es decir, es aquel que, a diferencia de los autómatas finitos deterministas, posee al menos un estado, tal que para un símbolo del alfabeto, existe más de una transición posible. En palabras más simples, en un NFA tenemos varios caminos posibles dada una condición. Un Autómata Finito No Determinista consta de:

1. Un conjunto finito de estados, a menudo designado como Q .
2. Un conjunto finito de símbolos de entrada, a menudo designado como Σ (sigma).
3. Una función de transición que toma como argumentos un estado y un símbolo de entrada y devuelve un estado. La función de transición se designa habitualmente como δ o Δ (delta).
4. Un estado inicial, uno de los estados de Q .
5. Un conjunto de estados finales o de aceptación F . El conjunto F es un subconjunto de Q .



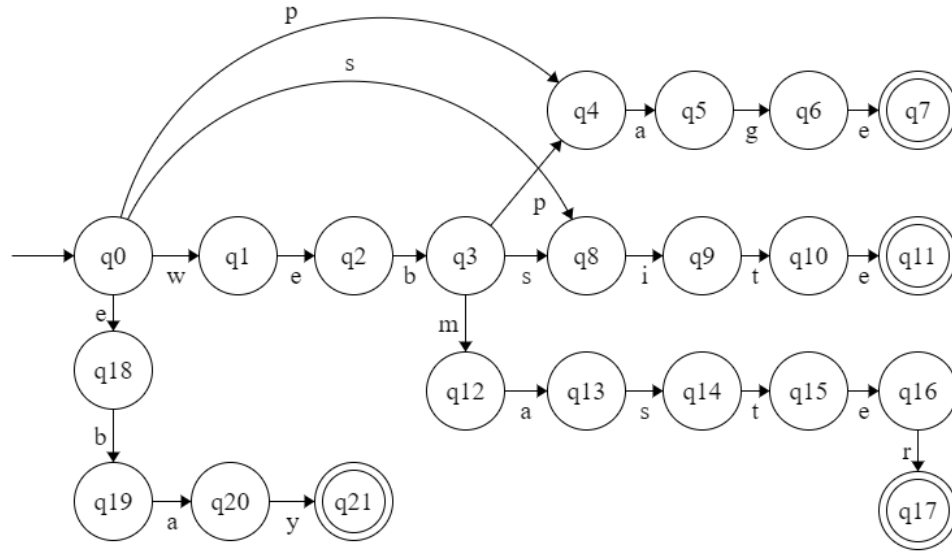
Representación gráfica de un NFA

Para resolver nuestro ejercicio, tendremos que convertir un NFA a DFA. Los AFN son útiles como herramienta teórica: facilitan el diseño y suelen ser más “económicos” (menor número de estados, menor número de transiciones) que los AFD. Pero, por desgracia, son herramientas teóricas: el mundo real es determinista, por lo tanto, para poder “verlos en funcionamiento” o bien se recurre a la simulación o bien se deben transformar previamente a AFD.

1. Se construye una tabla donde cada columna está etiquetada con un símbolo del alfabeto de entrada y cada fila se etiqueta con un conjunto de estados.
2. La primera fila se etiqueta con q_0 , estado inicial, y en cada entrada de la tabla $[q_0, si]$ se almacena $f(q_0, si) = p_1, \dots, p_n = P$.
3. Se etiqueta cada fila con cada uno de los conjuntos P que no tengan asociada una fila en la tabla (es decir, con cada uno de los conjuntos P que aparezcan por primera vez en la tabla) y se completa cada una de estas filas con el correspondiente $f(P, si)$.
4. Se realiza el paso (3) hasta que no haya en la tabla conjuntos P sin filas asociadas.
5. Se asocia a cada conjunto P que aparezca en la tabla un estado en el nuevo AFD y aquellos que tengan entre sus componentes algún estado final del AFN se considerarán estados finales en el AFD.

3 Desarrollo

En la implementación, primero planteamos un autómata que nos permita aceptar el lenguaje $L = [web, website, webpage, webmaster, page, site, ebay]$, por lo tanto tendrá un alfabeto determinado por $\Sigma = [w, e, b, s, i, t, p, a, g, m, r, y]$



Representación gráfica del NFA propuesto

	w	e	b	p	a	g	s	i	t	m	r	y
q0	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q1	q0,q1	q0,q18,q2	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q2	q0,q1	q0,q18	q0,q3	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q3	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0,q12	q0	q0
q4	q0,q1	q0,q18	q0	q0,q4	q0,q5	q0	q0,q8	q0	q0	q0	q0	q0
q5	q0,q1	q0,q18	q0	q0,q4	q0	q0,q6	q0,q8	q0	q0	q0	q0	q0
q6	q0,q1	q0,q18,q7	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q7	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q8	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0,q9	q0	q0	q0	q0
q9	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0,q10	q0	q0	q0
q10	q0,q1	q0,q18,q11	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q11	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q12	q0,q1	q0,q18	q0	q0,q4	q0,q13	q0	q0,q8	q0	q0	q0	q0	q0
q13	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8,q14	q0	q0	q0	q0	q0
q14	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0,q15	q0	q0	q0
q15	q0,q1	q0,q18,q16	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q16	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0,q17	q0
q17	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q18	q0,q1	q0,q18	q0,q19	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0
q19	q0,q1	q0,q18	q0	q0,q4	q0,q20	q0	q0,q8	q0	q0	q0	q0	q0
q20	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0,q21
q21	q0,q1	q0,q18	q0	q0,q4	q0	q0	q0,q8	q0	q0	q0	q0	q0

	w	e	b	p	a	g	s	i	t	m	r	y
q0	q01	q018	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q01	q01	q0182	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q018	q01	q018	q019	q04	q0	q0	q08	q0	q0	q0	q0	q0
q04	q01	q018	q0	q04	q05	q0	q08	q0	q0	q0	q0	q0
q08	q01	q018	q0	q04	q0	q0	q08	q09	q0	q0	q0	q0
q05	q01	q018	q0	q04	q0	q06	q08	q0	q0	q0	q0	q0
q06	q01	q0187	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q09	q01	q018	q0	q04	q0	q0	q08	q0	q010	q0	q0	q0
q010	q01	q01811	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q01811	q01	q018	q019	q04	q0	q0	q08	q0	q0	q0	q0	q0
q019	q01	q018	q0	q04	q020	q0	q08	q09	q0	q0	q0	q0
q020	q01	q018	q0	q04	q0	q0	q08	q0	q0	q0	q0	q021
q021	q01	q018	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q0182	q01	q018	q03	q04	q0	q0	q08	q0	q0	q0	q0	q0
q03	q01	q018	q0	q04	q0	q0	q08	q0	q0	q012	q0	q0
q012	q01	q018	q0	q04	q013	q0	q08	q0	q0	q0	q0	q0
q013	q01	q018	q0	q04	q0	q0	q0814	q0	q0	q0	q0	q0
q0814	q01	q018	q0	q04	q0	q0	q08	q09	q015	q0	q0	q0
q015	q01	q01816	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q01816	q01	q018	q019	q04	q0	q0	q08	q0	q0	q0	q017	q1
q017	q01	q018	q0	q04	q0	q0	q08	q0	q0	q0	q0	q0
q0187	q01	q018	q019	q04	q0	q0	q08	q0	q0	q0	q0	q0

		w	e	b	p	a	g	s	i	t	m	r	y
q0	q0	q1	q2	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0
q1	q01	q1	q5	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0
q2	q018	q1	q2	q6	q3	q0	q0	q4	q0	q0	q0	q0	q0
q3	q04	q1	q2	q0	q3	q7	q0	q4	q0	q0	q0	q0	q0
q4	q08	q1	q2	q0	q3	q0	q0	q4	q8	q0	q0	q0	q0
q5	q0182	q1	q2	q9	q3	q0	q0	q4	q0	q0	q0	q0	q0
q6	q019	q1	q2	q0	q3	q10	q0	q4	q8	q0	q0	q0	q0
q7	q05	q1	q2	q0	q3	q0	q11	q4	q0	q0	q0	q0	q0
q8	q09	q1	q2	q0	q3	q0	q0	q4	q0	q12	q0	q0	q0
q9	q03	q1	q2	q0	q3	q0	q0	q4	q0	q0	q13	q0	q0
q10	q020	q1	q2	q0	q3	q0	q0	q4	q0	q0	q0	q0	q14
q11	q06	q1	q15	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0
q12	q010	q1	q16	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0
q13	q012	q1	q2	q0	q3	q17	q0	q4	q0	q0	q0	q0	q0
q14	q021	q1	q2	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0
q15	q0187	q1	q2	q6	q3	q0	q0	q4	q0	q0	q0	q0	q0
q16	q01811	q1	q2	q6	q3	q0	q0	q4	q0	q0	q0	q0	q0
q17	q013	q1	q2	q0	q3	q0	q0	q18	q0	q0	q0	q0	q0
q18	q0814	q1	q2	q0	q3	q0	q0	q4	q8	q19	q0	q0	q0
q19	q015	q1	q20	q0	q3		q0	q4	q0	q0	q0	q0	q0
q20	q01816	q1	q2	q6	q3	q0	q0	q4	q0	q0	q0	q21	q1
q21	q017	q1	q2	q0	q3	q0	q0	q4	q0	q0	q0	q0	q0

```

1 #AFN a AFD (WebEbay)
2
3 from atexit import register
4 from tkinter import*
5
6 leer = open("programa 4/prog4.txt", "r")
7 registro = open("programa 4/registro.txt", "w")
8 posicion = open("programa 4/posicion.txt", "w")
9 prevstate = 0
10 auxweb1 = 0
11 auxweb2 = 0
12 web = [ ] #
13 website = [ ] #
14 webpage = [ ] #
15 website = [ ] #
16 webmaster = [ ] #
17 site = [ ] #
18 page = [ ] #
19 ebay = [ ] #
20
21
22 contador = 0

```

```

23 estado = 0
24
25 texto = leer.read()
26
27 for letra in texto.lower():
28
29     if letra == "\n": letra = "\n"
30     contador += 1
31
32     if (estado == 0):
33
34         if (letra == "w"):
35             estado = 1
36         elif (letra == "e"):
37             estado = 2
38         elif (letra == "p"):
39             estado = 3
40         elif (letra == "s"):
41             estado = 4
42         else:
43             estado = 0
44
45         registro.write("(q0,\""+letra+"\",q"+str(estado)+")\n")
46         continue
47
48     if (estado == 1):
49
50         if (letra == "w"):
51             estado = 1
52         elif (letra == "e"):
53             estado = 5
54         elif (letra == "p"):
55             estado = 3
56         elif (letra == "s"):
57             estado = 4
58         else:
59             estado = 0
60
61         registro.write("(q1,\""+letra+"\",q"+str(estado)+")\n")
62         continue
63
64     if (estado == 2):
65
66         if (letra == "w"):
67             estado = 1
68         elif (letra == "e"):
69             estado = 2
70         elif (letra == "p"):
71             estado = 3
72         elif (letra == "s"):
73             estado = 4
74         elif (letra == "b"):
75             estado = 6
76         else:
77             estado = 0
78
79         registro.write("(q2,\""+letra+"\",q"+str(estado)+")\n")

```

```

80         continue
81
82     if (estado == 3):
83         auxweb1 = 0
84         if (letra == "w"):
85             estado = 1
86         elif (letra == "e"):
87             estado = 2
88         elif (letra == "p"):
89             estado = 3
90         elif (letra == "s"):
91             estado = 4
92         elif (letra == "a"):
93             estado = 7
94             if prevstate == 9:
95                 #print("prev = ",prevstate)
96                 auxweb1 = 1
97                 auxweb1 += 1
98                 prevstate = 0
99         else:
100             estado = 0
101     registro.write("(q3,\""+letra+"\",q"+str(estado)+")\n")
102     continue
103
104
105     if (estado == 4):
106         auxweb2 = 0
107         if (letra == "w"):
108             estado = 1
109         elif (letra == "e"):
110             estado = 2
111         elif (letra == "p"):
112             estado = 3
113         elif (letra == "s"):
114             estado = 4
115         elif (letra == "i"):
116             estado = 8
117             print("prev = ",prevstate)
118             if prevstate == 9:
119                 auxweb2 = 1
120                 auxweb2 += 1
121                 prevstate = 0
122             print("auxweb2 = ",auxweb2)
123         else:
124             estado = 0
125
126     registro.write("(q4,\""+letra+"\",q"+str(estado)+")\n")
127     continue
128
129
130     if (estado == 5):
131         if (letra == "w"):
132             estado = 1
133         elif (letra == "e"):
134             estado = 2
135         elif (letra == "p"):
136             estado = 3

```

```

137     elif (letra == "s"):
138         estado = 4
139     elif(letra == "b"):
140         estado = 9
141     else:
142         estado = 0
143
144     registro.write("(q5,\""+letra+"\",q"+str(estado)+")\n")
145     continue
146
147
148 if (estado == 6):
149     if (letra == "w"):
150         estado = 1
151     elif (letra == "e"):
152         estado = 2
153     elif (letra == "p"):
154         estado = 3
155     elif (letra == "s"):
156         estado = 4
157     elif(letra == "a"):
158         estado = 10
159     elif(letra == "i"):
160         estado = 8
161     else:
162         estado = 0
163
164     registro.write("(q6,\""+letra+"\",q"+str(estado)+")\n")
165     continue
166
167 if (estado == 7):
168     if (letra == "w"):
169         estado = 1
170     elif (letra == "e"):
171         estado = 2
172     elif (letra == "p"):
173         estado = 3
174     elif (letra == "s"):
175         estado = 4
176     elif(letra == "g"):
177         estado = 11
178         auxweb1 += 1
179     else:
180         estado = 0
181
182     registro.write("(q7,\""+letra+"\",q"+str(estado)+")\n")
183     continue
184
185 if (estado == 8):
186     if (letra == "w"):
187         estado = 1
188     elif (letra == "e"):
189         estado = 2
190     elif (letra == "p"):
191         estado = 3
192     elif (letra == "s"):
193         estado = 4

```



```

194         elif(letra == "t"):
195             estado = 12
196             auxweb2 += 1
197         else:
198             estado = 0
199
200         registro.write("(q8,\""+letra+"\",q"+str(estado)+")\n")
201         continue
202
203     if (estado == 9): #web
204         web.append(contador-3)
205         print("letra = ",letra)
206
207         if (letra == "w"):
208             estado = 1
209         elif (letra == "e"):
210             estado = 2
211         elif (letra == "p"):
212             print("estado3")
213             prevstate = estado
214             estado = 3
215         elif (letra == "s"):
216             print("estado4")
217             prevstate = estado
218             estado = 4
219         elif(letra == "m"):
220             estado = 13
221         else:
222             estado = 0
223
224         registro.write("(q9,\""+letra+"\",q"+str(estado)+")\n")
225         continue
226
227     if (estado == 10):
228         if (letra == "w"):
229             estado = 1
230         elif (letra == "e"):
231             estado = 2
232         elif (letra == "p"):
233             estado = 3
234         elif (letra == "s"):
235             estado = 4
236         elif(letra == "y"):
237             estado = 14
238         else:
239             estado = 0
240
241         registro.write("(q10,\""+letra+"\",q"+str(estado)+")\n")
242         continue
243
244     if (estado == 11):
245         if (letra == "w"):
246             estado = 1
247         elif (letra == "e"):
248             estado = 15
249             auxweb1 += 1
250         elif (letra == "p"):

```

```

251         estado = 3
252     elif (letra == "s"):
253         estado = 4
254     else:
255         estado = 0
256
257     registro.write("(q11,\""+letra+"\",q"+str(estado)+")\n")
258     continue
259
260 if (estado == 12):
261
262     if (letra == "w"):
263         estado = 1
264     elif (letra == "e"):
265         estado = 16
266         auxweb2 += 1
267     elif (letra == "p"):
268         estado = 3
269     elif (letra == "s"):
270         estado = 4
271     else:
272         estado = 0
273
274     registro.write("(q12,\""+letra+"\",q"+str(estado)+")\n")
275     continue
276
277 if (estado == 13):
278     if (letra == "w"):
279         estado = 1
280     elif (letra == "e"):
281         estado = 2
282     elif (letra == "p"):
283         estado = 3
284     elif (letra == "s"):
285         estado = 4
286     elif (letra == "a"):
287         estado = 17
288     else:
289         estado = 0
290
291     registro.write("(q13,\""+letra+"\",q"+str(estado)+")\n")
292     continue
293
294 if (estado == 14): #ebay
295
296     ebay.append(contador-4)
297
298     if (letra == "w"):
299         estado = 1
300     elif (letra == "e"):
301         estado = 2
302     elif (letra == "p"):
303         estado = 3
304     elif (letra == "s"):
305         estado = 4
306     else:
307         estado = 0

```

```

308
309     registro.write("(q14,\""+letra+"\","+str(estado)+")\n")
310     continue
311
312 if (estado == 15): #page/webpage
313
314     if auxweb1 == 4:
315         webpage.append(contador-7)
316         page.append(contador-4)
317
318     if (letra == "w"):
319         estado = 1
320     elif (letra == "e"):
321         estado = 2
322     elif (letra == "p"):
323         estado = 3
324     elif (letra == "s"):
325         estado = 4
326     elif (letra == "b"):
327         estado = 6
328     else:
329         estado = 0
330
331     registro.write("(q15,\""+letra+"\","+str(estado)+")\n")
332     auxweb1 = 0
333     continue
334
335 if (estado == 16): #site/website
336     print("auxweb2",auxweb2)
337     if auxweb2 == 4:
338         website.append(contador-7)
339         site.append(contador-4)
340
341     if (letra == "w"):
342         estado = 1
343     elif (letra == "e"):
344         estado = 2
345     elif (letra == "p"):
346         estado = 3
347     elif (letra == "s"):
348         estado = 4
349     elif (letra == "b"):
350         estado = 6
351     else:
352         estado = 0
353
354     registro.write("(q16,\""+letra+"\","+str(estado)+")\n")
355     auxweb2 = 0
356     continue
357
358 if (estado == 17):
359     if (letra == "w"):
360         estado = 1
361     elif (letra == "e"):
362         estado = 2
363     elif (letra == "p"):
364         estado = 3

```

```

365         elif (letra == "s"):
366             estado = 18
367         else:
368             estado = 0
369
370         registro.write("(q17,\""+letra+"\",q"+str(estado)+")\n")
371         continue
372
373     if (estado == 18):
374         if (letra == "w"):
375             estado = 1
376         elif (letra == "e"):
377             estado = 2
378         elif (letra == "p"):
379             estado = 3
380         elif (letra == "s"):
381             estado = 4
382         elif(letra == "t"):
383             estado = 19
384         elif(letra == "i"):
385             estado = 8
386         else:
387             estado = 0
388
389         registro.write("(q18,\""+letra+"\",q"+str(estado)+")\n")
390         continue
391
392     if (estado == 19):
393         if (letra == "w"):
394             estado = 1
395         elif (letra == "e"):
396             estado = 20
397         elif (letra == "p"):
398             estado = 3
399         elif (letra == "s"):
400             estado = 4
401         else:
402             estado = 0
403
404         registro.write("(q19,\""+letra+"\",q"+str(estado)+")\n")
405         continue
406
407     if (estado == 20):
408         if (letra == "w"):
409             estado = 1
410         elif (letra == "e"):
411             estado = 2
412         elif (letra == "p"):
413             estado = 3
414         elif (letra == "s"):
415             estado = 4
416         elif(letra == "b"):
417             estado = 6
418         elif(letra == "r"):
419             estado = 21
420         else:
421             estado = 0

```

```

422     registro.write("(q20,\""+letra+"\",q"+str(estado)+")\n")
423     continue
424
425
426     if (estado == 21): #webmaster
427         if (letra == "w"):
428             estado = 1
429         elif (letra == "e"):
430             estado = 2
431         elif (letra == "p"):
432             estado = 3
433         elif (letra == "s"):
434             estado = 4
435         else:
436             estado = 0
437         webmaster.append(contador-9)
438         registro.write("(q21,\""+letra+"\",q"+str(estado)+")\n")
439         continue
440
441     posicion.write("                REGISTRO DE APARICIONES\n")
442     posicion.write("Web = "+ str(len(web))+ "\tPosiciones = "+str(web)+"\n")
443     posicion.write("ebay = "+ str(len(eybay))+ "\tPosiciones = "+str(eybay)+ "\n")
444     posicion.write("page = "+ str(len(page))+ "\tPosiciones = "+str(page)+ "\n")
445     posicion.write("site = "+ str(len(site))+ "\tPosiciones = "+str(site)+ "\n")
446     posicion.write("webpage = "+ str(len(webpage))+ "\tPosiciones = "+str(webpage)+ "\n")
447     posicion.write("Website = "+ str(len(website))+ "\tPosiciones = "+str(website)+ "\n")
448     posicion.write("Webmaster = "+ str(len(webmaster))+ "\tPosiciones = "+str(webmaster)+ "\n")
449
450
451
452 leer.close()
453 posicion.close()
454 registro.close()
455 }

```

Algoritmo 1: Implementación de DFA

REGISTRO DE APARICIONES	
Web - 9	Posiciones = [16, 466, 623, 817, 986, 972, 1425, 1495, 1867]
ebay - 29	Posiciones = [20, 231, 332, 435, 461, 610, 701, 776, 849, 901, 988, 1074, 1213, 1301, 1398, 1490, 1753, 1791, 1862, 2185, 2751, 2968, 3021, 3165, 3373, 3458, 3636,
page - 11	Posiciones = [820, 1009, 1334, 1520, 1711, 1816, 1894, 2001, 2245, 2289, 2870]
site - 7	Posiciones = [469, 909, 975, 1428, 1498, 1870, 2793]
webpage - 1	Posiciones = [817]
Website - 6	Posiciones = [466, 986, 972, 1425, 1495, 1867]
Webmaster - 1	Posiciones = [623]

1	(q0,"p",q3)
2	(q3,"r",q0)
3	(q0,"o",q0)
4	(q0,"m",q0)
5	(q0,"o",q0)
6	(q0,"t",q0)
7	(q0,"i",q0)
8	(q0,"n",q0)
9	(q0,"g",q0)
10	(q0,"",q0)
11	(q0,"y",q0)
12	(q0,"o",q0)
13	(q0,"u",q0)
14	(q0,"r",q0)
15	(q0,"",q0)
16	(q0,"w",q1)
17	(q1,"e",q5)
18	(q5,"b",q9)
19	(q9,"",q0)
20	(q0,"b",q0)
21	(q0,"i",q0)
22	(q0,"z",q0)
23	(q0,"",q0)
24	(q0,"w",q1)
25	(q1,"i",q0)
26	(q0,"t",q0)
27	(q0,"h",q0)
28	(q0,"",q0)
29	(q0,"e",q2)
30	(q2,"b",q6)
31	(q6,"a",q10)
32	(q10,"y",q14)
33	(q14,"n",q0)
34	(q0,"a",q0)
35	(q0,"r",q0)
36	(q0,"e",q2)
37	(q2,"",q0)
38	(q0,"y",q0)
39	(q0,"o",q0)
40	(q0,"u",q0)
41	(q0,"",q0)
42	(q0,"p",q3)
43	(q3,"r",q0)
44	(q0,"o",q0)
45	(q0,"m",q0)
46	(q0,"o",q0)
47	(q0,"t",q0)
48	(q0,"i",q0)
49	(q0,"n",q0)

4 Conclusión

Gracias a este programa, fuimos capaces de mejorar y comprender al completo tanto los NFA's como los DFA's. Del mismo modo, podemos ver una implementación de la conversión de estos últimos. Me parece un ejercicio muy interesante ya que hacemos uso de recursos variados, desde lógica de programación, hasta modelos matemáticos. El mayor reto durante este tipo de tareas, es generar un código que obedezca a nuestro grafo planteado, ya que en un lenguaje de programación, no todo funciona como el desarrollador quiere.

References

- [1] Como se transforma un AFN en un AFD? (s. f.). Teoría de Automatas y Lenguajes Formales. Recuperado 5 de junio de 2022, de <http://repositori.uji.es/xmlui/bitstream/handle/10234/5875/bolAuto1.pdf?sequence=1>
- [2] NIVERSIDAD DE MAGALLANES FACULTAD DE INGENIERÍA DEPARTAMENTO DE COMPUTACIÓN. (2004, 31 junio). Automatas Finitos No Deterministas. AUTÓMATAS FINITOS NO DETERMINISTAS. Recuperado 5 de junio de 2022, de <https://kataix.umag.cl/jaguila/Compilers/T04AFN.pdf>
- [3] UTÓMATA FINITO NO DETERMINISTA — Autómatas Wiki — Fandom. (2009). Autómatas Wiki. Recuperado 5 de junio de 2022, de <https://automatas.fandom.com/es/wiki/AUTC393MATAFINITONODETERMINISTA>