Teoría de Autómatas y Lenguajes Formales

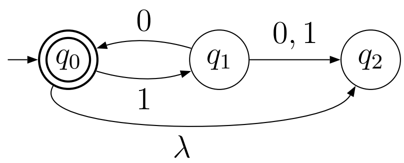
Ingeniería Técnica en Informática de Sistemas

Hoja de Problemas 5

Autómatas Finitos No Deterministas

Nivel del ejercicio : ($\star$) básico, ($\clubsuit$) medio, ($\spadesuit$) avanzado.

1. ($\star$) ¿Cuál es el lenguaje reconocido por el autómata siguiente?

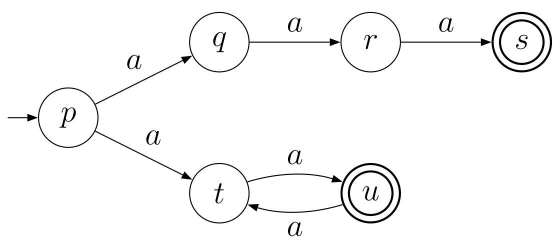


Solución:

El lenguaje que reconoce este autómata es un lenguaje binario donde las palabras aceptadas son de la siguiente manera:

$$L = \{(10)^n \mid n \geq 0\}$$

2. ($\star$) Dado el siguiente autómata:



(a) Indica el lenguaje L reconocido.

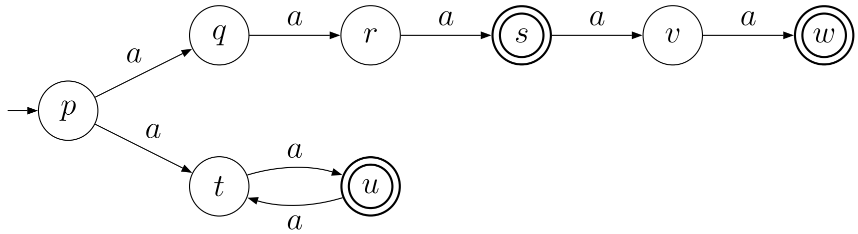
Solución:

El lenguaje que reconoce este autómata es el siguiente:

$$L = \{w \in \{a\}^* \mid n_a(w) \; mod \; 2 = 0\}$$

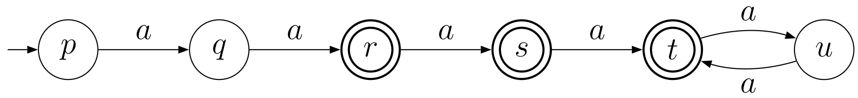
(b) Construye un AFND que genere el lenguaje $L \cup \{a^5\}$.

Solución:

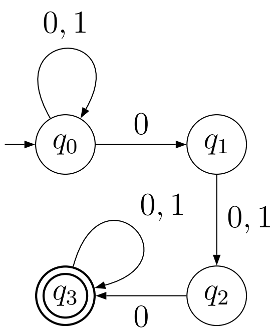


(c) Encuentra un AFD equivalente.

Solución:



3. ($\star$) ¿ Cuál es el lenguaje reconocido por el autómata siguiente?



Solución:

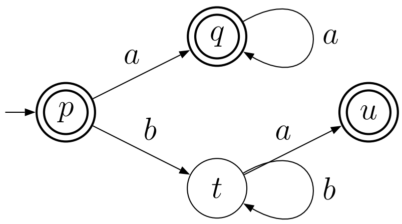
El lenguaje que reconoce este autómata es un lenguaje binario donde las palabras aceptadas tienen, al menos, dos ceros separados por un carácter:

$$L = \{x0y0z \mid x, z \in \{0,1\}^*,\ y \in \{0,1\}\}$$

4. ($\clubsuit$) Construye un AFND con cuatro estados que reconozca el lenguaje L siguiente:

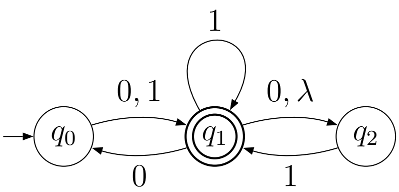
$$L = \{a^n \mid n \geq 0\} \cup \{b^n a \mid n \geq 1\}.$$

Solución:



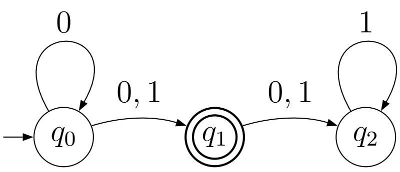
5. (♣) Para cada uno de los autómatas finitos no deterministas siguientes, calcula un autómata finito determinista equivalente mínimo:

(a)



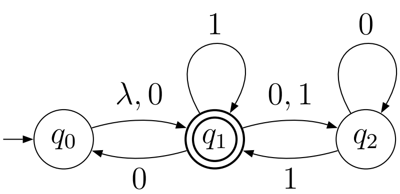
Solución:

(b)

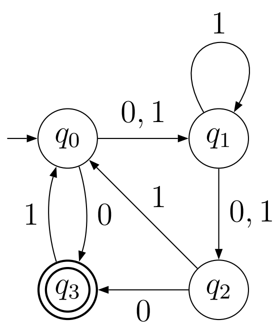


Solución:

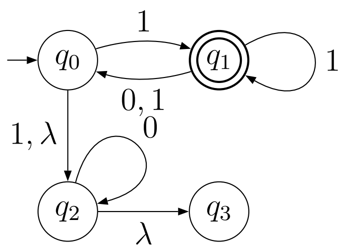
(c)



Solución:

(d) 

Solución:

(e) 

Solución:

6. (♣) Para cada uno de los autómatas finitos no deterministas siguientes, calcula un AFD equivalente mínimo:

   (a) $AFND_1 = (\{a, b\}, \{p, q, r, s\}, f_1, p, \{s\})$

f_1	a	b	λ
→p	q,s	p	q,r
q		q,r	r
r		p,s	q
*s	s	q,r,s	

Solución:

f_1	a	b	λ
→p	q,s	p	q,r
q		q,r	r
r		p,s	q
*s	s	q,r,s	

Primero debemos obtener un AFD a partir de este AFND. En los apuntes existen dos métodos a aplicar. El primero consiste en generar una tabla de transiciones donde se tengan en cuenta todas las posibles partes de Q, o sea, 2^Q. Este método puede dar lugar a estados inaccesibles que luego deberían ser eliminados. Como alternativa podemos utilizar el otro método que consiste en empezar a construir la tabla de transición a partir del estado

$CLAUS_\lambda(q_0) = q'_0$, y definiendo a continuación todos los estados $f'(q_0, a)$ que vayan surgiendo, hasta que no existan nuevos estados nuevos. Aquí seguiremos este método.

Primero vemos cual será el nuevo estado inicial:

$$CLAUS_\lambda(q_0) = \{p, q, r\}$$

Ahora deberemos ir rellenando la tabla de transiciones para cada uno de los símbolos del alfabeto (excluyendo λ) para este nuevo estado y vamos añadiendo los nuevos estados que se van creando y rellenando sus correspondientes entradas en la tabla. De esta forma, llegamos al AFD equivalente:

f'_1	a	b
→{p,q,r}	{q,r,s}	{p,q,r,s}
*{q,r,s}	s	{p,q,r,s}
*{p,q,r,s}	{q,r,s}	{p,q,r,s}
*s	s	{q,r,s}

Hacemos un cambio de nombres para que quede más limpio, quedando:

f'_1	a	b
→q_a	q_b	q_c
*q_b	q_d	q_c
*q_c	q_b	q_c
*q_d	q_d	q_b

Ahora minimizamos este autómata:

$$Q/E_0 = \{\underbrace{\{q_a\}}_{p}, \underbrace{\{q_b, q_c, q_d\}}_{c}\} = Q/E_1 = Q/E$$

Renombrando nos queda la siguiente tabla de transición del autómata mínimo:

f'_1	a	b
→p	c	c
*c	c	c

(b) $AFND_2 = (\{a, b\}, \{p, q, r, s, t, u, v\}, f_2, p, \{v\})$

f_2	a	b
→p	q,r	p
q	q,r	s,t,u
r		p,v
s	r,u	
t		
u	s,t	v
*v	u,s,t	v

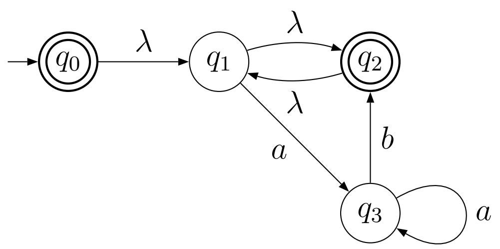
Solución:

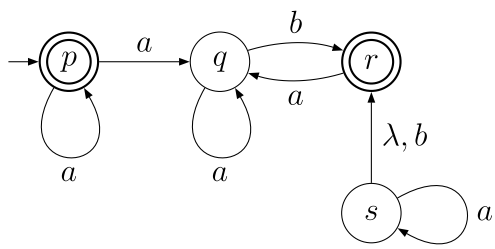
(c) $AFND_3 = (\{a, b, c\}, \{p, q, r, s, t, u, v\}, f_3, p, \{v\})$

f_3	a	b	c	λ
→p				q,t
q		r,s		r,s
r				q,u
s	t,p		u	
t		v		q
u	s,q		v	s
*v				r

Solución:

7. (♣) Dados los autómatas siguientes:





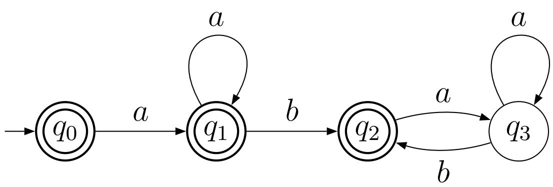
(a) Determina si son equivalentes (calculando autómatas mínimos).

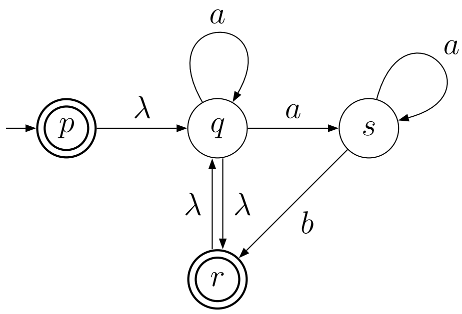
Solución:

(b) Obtén gramáticas lineales derechas que generen los lenguajes reconocidos por dichos autómatas.

Solución:

8. (♣) Dados los autómatas siguientes:





(a) Determina si son equivalentes (calculando autómatas mínimos).

> **Solución:**

(b) Obtén gramáticas lineales derechas que generen los lenguajes reconocidos por dichos autómatas.

> **Solución:**

9. (♣) Construye un AFD mínimo que reconozca el lenguaje generado por la gramática con las producciones siguientes:

$$\begin{array}{lcl} A & ::= & 0B \mid \lambda \\ B & ::= & 1C \mid 1 \\ C & ::= & 0B \end{array}$$

> **Solución:**

10. (♣) Construye un AFD mínimo que reconozca el lenguaje generado por la gramática con las producciones siguientes:

$$\begin{array}{lcl} S & ::= & bS \mid aA \mid \lambda \\ A & ::= & aA \mid bB \\ B & ::= & bS \mid \lambda \end{array}$$

> **Solución:**
>
> Lo primero que debemos hacer es limpiar la gramática de reglas reductoras que no sean del tipo $S ::= \lambda$.
>
> $$S \; ::= \; bS \mid aA \mid \lambda$$

$$\begin{array}{rcl} A & ::= & aA \mid bB \mid b \\ B & ::= & bS \end{array}$$

Generamos el AFND correspondiente a esta gramática regular lineal por la derecha:

f	a	b
→*S	A	S
A	A	{B,F}
*F	∅	∅
B	∅	S

El AFD correspondiente (y mínimo a su vez) es:

f'	a	b
→*S	A	S
A	A	{B,F}
*{B,F}	∅	S
∅	∅	∅

Renombrando los estados se obtiene el AFD mínimo final:

f'	a	b
→*q_0	q_1	q_0
q_1	q_1	q_2
*q_2	q_3	q_0
q_3	q_3	q_3

11. (♣) Construye un AFD mínimo que reconozca el lenguaje generado por la gramática con las producciones siguientes:

$$\begin{array}{rcl} S & ::= & bR_1 \mid cR_2 \mid a \mid b \\ R_1 & ::= & aR_1 \mid a \\ R_2 & ::= & cR_2 \mid a \end{array}$$

Solución: