Universidad Nacional Autónoma de México

Facultad de Ciencias

Automatas y Lenguajes Formales

Tarea 01

Arriaga Santana Estela Monserrat
Castañon Maldonado Carlos Emilio
Fernández Blancas Melissa Lizbeth

1 Sean $L_1, L_2 \in \mathcal{P}(\Sigma^*)$ lenguajes, demostrar que la concatenación cumple las siguientes propiedades.:

(a) $L_1 \cdot \{\epsilon\} = L_1 = \{\epsilon\} \cdot L_1$
Sea L_1 un lenguaje sobre el alfabeto Σ, entonces tenemos que:

$$\begin{aligned} L_1 \cdot \{\epsilon\} &= \{w_1 | w_1 \in L_1\} \cdot \{\epsilon\} \quad \text{Por definición de } L_1 \\ &= \{w_1 \cdot w_2 | w_1 \in L_1, w_2 \in \{\epsilon\}\} \quad \text{Por definicion de concatenación} \\ &= \{w_1 \cdot \epsilon | w_1 \in L_1\} \ \text{ Como } w_2 \in \{\epsilon\}, \text{ entonces } w_2 = \epsilon \\ &= \{w_1 | w_1 \in L_1\} = L_1 \quad \text{Por propiedad de neutro de las cadenas} \\ &= \{\epsilon \cdot w_1 | w_1 \in L_1\}; \ \text{ Por neutro de cadenas} \\ &= \{w_2 \cdot w_1 | w_1 \in L_1, w_2 \in \{\epsilon\}\}; \ \ w_2 = \epsilon \\ &= \{\epsilon\} \cdot \{w_1 | w_1 \in L_1\}; \ \text{ Por definición de concatenación} \\ &= \{\epsilon\} \cdot L_1; \ \text{ Por definición de } L_1 \end{aligned}$$

(b) $L_1 \cdot (L_2 \cdot L_3) = (L_1 \cdot L_2) \cdot L_3$
Sean L_1, L_2, L_3 lenguajes sobre Σ^*, entonces:

$$\begin{aligned} L_1 \cdot (L_2 \cdot L_3) &= L_1 \cdot \{w_2 w3 | w_2 \in L_2, w_3 \in L_3\} \quad \text{Definición de concatenación} \\ &= \{w_1(w_2 w_3) | w_1 \in L_1, w_2 \in L_2, w_3 \in L_3\} \quad \text{Definición de concatenación} \\ &= \{(w_1 w_2) w_3 | w_1 \in L_1, w_2 \in L_2, w_3 \in L_3\} \quad \text{Propiedad de asociatividad de las cadenas} \\ &= \{w_1 w_2 | w_1 \in L_1, w_2 \in L_{L2}\} \cdot L_3 \quad \text{Por definición de concatenación} \\ &= (L_1 \cdot L_2) \cdot L_3 \ \text{ Por definición de concatenación} \end{aligned}$$

Haciendo la demostración por inducción, entonces:
Sea $L = L_1(L_2 L_3)$ y $w = w_1(w_2 w_3)$ con $w_1 \in L_1$, $w_2 \in L_2$ y $w_3 \in L_3$, procedemos, inducción sobre w_1.
Caso base: Si $w_1 = \epsilon$, entonces

$$\begin{aligned} w_1(w_2 w_3) &= \epsilon(w_2 w_3) \\ &= w_2 w_3 \quad \text{Por neutro en la concatenación de cadenas} \\ &= (w_2) w_2 \quad \text{Por asociatividad de cadenas en la concatenación} \\ &= (\epsilon w_2) w_3 \quad \text{Por neutro en la concatenación de cadenas} \\ &= (w_1 w_2) w_3 \quad \text{Pues } w_1 = \epsilon \end{aligned}$$

Hipótesis de inducción: Supongamos que $w_1(w_2 w_3) = (w_1 w_2) w_3$ con $w_1 = x$.
Paso inductivo: Por demostrar que $w_1(w_2 w_3) = (w_1 w_2) w_3$ con $w_1 = ax$, $a \in \Sigma$.

$$\begin{aligned} w_1(w_2 w_3) &= ax(w_2 w_3) \quad \text{Pues } w_1 = ax \\ &= a(x(w_2 w_3)) \quad \text{Por propiedad de asociatividad de las cadenas} \\ &= a((x w_2) w_3) \quad \text{Por hipótesis de inducción} \\ &= a(x w_2) w_3 \quad \text{Por concatenación} \\ &= (a x w_2) w_3 \quad \text{Por asociatividad} \\ &= (w_1 w_2) w_3 \quad \text{Por definición de } W_1 \end{aligned}$$

Por lo tanto $L_1 \cdot (L_2 \cdot L_3) = (L_1 \cdot L_2) \cdot L_3$.

2 Demostrar la distributividad de la unión y la concatenación para los lenguajes. Esto es:

$$L_1 \cdot (L_2 \cup L_3) = (L_1 \cdot L_2) \cup (L_1 \cdot L_3)$$

Recordando que los lenguajes son **conjuntos**, las operaciones entre lenguajes son las mismas operaciones que sus homologas en conjuntos teniendo de esta forma que la unión esta dada como:
$A \cup B = \{w | w \in A \quad \vee \quad w \in B\}$
Además de que la concatenación esta definida como:
$A \cdot B = \{wx | w \in A \quad \wedge \quad x \in B\}$
Por lo que procedemos a demostrar la distributividad de la unión y la concatenación para los lenguajes:
Sean L_1, L_2, L_3 lenguajes sobre el alfabeto Σ, con $x, w \in L_1 \cdot (L_2 \cup L_3)$ entonces tenemos que:

$$\begin{aligned}
&L_1 \cdot (L_2 \cup L_3) = (L_1 \cdot L_2) \cup (L_1 \cdot L_3) \\
&L_1 \cdot (L_2 \cup L_3) \iff w \in L_2 \cup L_3 \wedge x \in L_1 \\
&\iff (w \in L_2 \vee w \in L_3) \wedge x \in L_1 \\
&\iff (w \in L_2 \wedge x \in L_1) \vee (w \in L_3 \wedge x \in L_1) \quad \text{Por Ley Distributiva del } \wedge \text{ sobre el } \vee \\
&\iff (w \in L_1 \wedge x \in L_2) \vee (w \in L_1 \wedge x \in L_3) \quad \text{Por Asociatividad} \\
&\iff (w, x) \in L_1 \cdot L_2 \vee (w, x) \in L_1 \cdot L_3 \\
&\iff (w, x) \in (L_1 \cdot L_2) \cup (L_1 \cdot L_3)
\end{aligned}$$

3 Defínase la gramática $G = (\Sigma, \Delta, S, R)$ donde $\Sigma = \{0, 1\}$, $\Delta = \{S\}$, con S símbolo inicial y las reglas:

$$S \leftarrow 0S0|1S1|0|1|\epsilon$$

Demostrar que $G \models w$ si y sólo si $\exists x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0{+}1$(G genera el lenguaje de palíndromos).

- Por demostrar que si existe $x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0 + 1$, entonces $G \models w$
  Usaremos inducción sobre la forma de estructura de las cadenas.
  Paso base:
  - Sea $w = \epsilon$, sabemos que $\epsilon \in \Sigma^*$ y la gramática G tiene la regla de producción $S \rightarrow \epsilon$, por lo que $S \Rightarrow_G \epsilon$ y $G \models w$.
  - Sea $w = 0$, sabemos que $0 \in \Sigma$ y la gramática G tiene la regla de producción $S \rightarrow 0$, , por lo que $S \Rightarrow_G 0$ y $G \models w$.
  - Sea $w = 0$, sabemos que $1 \in \Sigma$ y la gramática G tiene la regla de producción $S \rightarrow 1$, , por lo que $S \Rightarrow_G 1$ y $G \models w$.

  Hipótesis de inducción: Supongamos que $w = x \cdot x^R$, $w \in \Sigma^*$ y que $S \Rightarrow_G^* w$.
  Paso inductivo: Por demostrar que $G \Rightarrow_G^* w$ con $w = axa$, $x \in \Sigma^*$, $a \in Sigma$, $x = y \cdot y^R$, $y \in \Sigma^*$.
  Por las reglas de producción de G y aplicando la hipótesis de inducción, tenemos dos casos:
  - $S \Rightarrow_G 0S0 \Rightarrow_G^* 0x0$, por lo que $G \models w$.
  - $S \Rightarrow_G 1S1 \Rightarrow_G^* 1x1$, por lo que $G \models w$.

  Por lo tanto, si existe $x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0 + 1$, entonces $G \models w$

- Por demostrar que si $G \models w$, entonces existe $x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0 + 1$.
  Usaremos inducción sobre las derivaciones de la gramática G
  Paso base: Sea $S \Rightarrow_G w$, es decir, la derivación se hace en un paso. La gramática G tiene tres producciones en un paso:
  - Sea $S \Rightarrow_G \epsilon$, entonces $w = \epsilon$ y $\epsilon \in \Sigma^*$, por lo que $w \in L$.
  - Sea $S \Rightarrow_G 0$, entonces $w = 0$ y $0 \in \Sigma$, por lo que $w \in L$.
  - Sea $S \Rightarrow_G 1$, entonces $w = 1$ y $1 \in \Sigma$, por lo que $w \in L$.

Hipótesis de inducción: Supongamos que $S \Rightarrow_G^* x$ en n pasos y que $x = y \cdot y^R$, con $y \in \Sigma^*$.
Paso inductivo: Por demostrar que $G \Rightarrow_G^* w = x \cdot x^R$ en $n+1$ pasos.
La gramática G tiene dos producciones que derivan en más de un paso: $S \to 0S0$ y $S \to 1S1$.
Como $S \Rightarrow_G^* w$, tenemos dos casos:

- Sea $w = 0x0$, entonces $S \Rightarrow_G 0S0 \Rightarrow_G^* 0x0$ y por hipótesis de inducción sabemos que la derivación $S \Rightarrow_G^* x$ sucede en n pasos y que $x = y \cdot y^R$ con $y \in \Sigma^*$, por lo que $S \Rightarrow_G^* 0x0$ se produce en $n+1$ pasos y $w = 0 \cdot y \cdot y^R \cdot 0 = (0 \cdot y) \cdot (y^R \cdot 0)$, por lo que podemos decir que $w = z \cdot z^R$ con $z = 0y$.

- Sea $w = 1x1$, entonces $S \Rightarrow_G 1S1 \Rightarrow_G^* 1x1$ y por hipótesis de inducción sabemos que la derivación $S \Rightarrow_G^* x$ sucede en n pasos y que $x = y \cdot y^R$ con $y \in \Sigma^*$, por lo que $S \Rightarrow_G^* 1x1$ se produce en $n+1$ pasos y $w = 1 \cdot y \cdot y^R \cdot 1 = (1 \cdot y) \cdot (y^R \cdot 1)$, por lo que podemos decir que $w = z \cdot z^R$ con $z = 1y$.

Por lo tanto, si $G \models w$, entonces existe $x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0 + 1$.
Luego, $G \models w$ si y sólo si $\exists x \in \Sigma^*$ tal que $w = xx^R$ ó $w = 0 + 1$.

4 Demostrar que el reverso de un lenguaje regular L^R es regular.

Sea $L = [[R]]$ un lenguaje regular, haremos inducción sobre la expresión regular R que define al lenguaje.
Caso base:

- Si $L = [[\emptyset]]$, entonces $R = \emptyset$, $R^R = \emptyset$, por lo qye $L^R = [[\emptyset]]$, que por definición es regular.
- Si $L = [[\epsilon]]$, entonces $R = \{\epsilon\}$, $R^R = \{\epsilon\}$, por lo que $L^R = [[\epsilon]]$, que por definición es regular.
- Si $L = [[a]]$ con $a \in \Sigma$, entonces $R = \{a\}$, $R^R = \{a\}$, por lo que $L^R = [[a]]$, que por definición es regular.

Hipótesis de inducción: Supongamos que $L_1 = [[R_1]]$ y $L_2 = [[R_2]]$ son lenguajes regulares sobre Σ^*, que $L_1^R = [[R_1^R]]$, $L_2^R = [[R_2^R]]$ donde R_1^R y R_2^R son expresiones regulares.
Paso inductivo: Por demostrar que:

- $L^R = (L_1 \cup L_2)^R$ es regular. Tenemos que

$$\begin{aligned} L^R &= (L_1 \cup L_2)^R \\ &= ([[R_1]] \cup [[R_2]])^R \quad \text{Por definición de lenguaje regular} \\ &= [[R_1 + R_2]]^R \quad \text{Por definición de denotación de una expresión regular} \\ &= [[R_1^R + R_2^R]] \quad \text{Por definición de reversa y de unión} \end{aligned}$$

Como R_1^R y R_2^R son regulares, entonces $[[R_1^R + R_2^R]]$ también es regular por la definición de denotación de lenguajes regulares. Por lo tanto $L^R = (L_1 \cup L_2)^R$ es regular.

- $L^R = (L_1 \cdot L_2)^R$ es regular.

$$\begin{aligned} L^R &= (L_1 \cdot L_2)^R \\ &= ([[R_1]] \cdot [[R_2]])^R \quad \text{Por definición de lenguaje regular} \\ &= [[R_1 \cdot R_2]]^R \quad \text{Por definición de denotación de una expresión regular} \\ &= [[(R_1 \cdot R_2)^R]] \quad \text{Por definición de reversa} \\ &= [[R_2^R \cdot R_1^R]] \quad \text{Por definición de reversa} \end{aligned}$$

Como R_1^R y R_2^R son regulares, entonces $[[R_2^R \cdot R_1^R]]$ también es regular por la definición de denotación de lenguajes regulares. Por lo tanto $L^R = (L_1 \cdot L_2)^R$ es regular.

- $L^R = (L_1^*)^R$ es regular.

$$\begin{aligned} L^R &= (L^*)^R \\ &= ([[R_1]]^*)^R \quad \text{Por definición de lenguaje regular} \\ &= [[R_1^R]]^* \quad \text{Por definición de reversa y de denotación con la estrella de Kleene} \end{aligned}$$

Como R_1^R es regular, entonces $[[r_1^R]]^*$ también es regular por la definición de denotación de lenguajes regulares. Por lo tanto $L^R = (L_1^*)^R$ es regular.

Por lo tanto, el reverso de un lenguaje regular L^R es regular.

5 Demostrar que los lenguajes regulares son cerrados bajo la concatenación. Esto es $L_1 \cdot L_2 \cdot ... \cdot L_n$ es regular.

Recordando las propiedades de un lenguaje regular , veamos que estos deben cumplir ciertas características, entre ellas las siguientes.

Comenzamos demostrando que los lenguajes regulares son cerrados bajo la concatenación esto es :

$L_1 \cup L_2 \cup L_3 \cup ... \cup L_n$ es regular.

Lo cual procedemos a demostrar:

Base:

$\exists r_1, r_2$ regex tal que:

$$L_1 \cup L_2 = [[r_1]] \cup [[r_2]] = [[r_1 + r_2]]$$

Por definicion tendremos que $r_1 + r_2$ es una regex y $[[r_1 + r_2]]$ es un lenguaje regular, por lo tanto $L_1 \cup L_2$ es regular.

Hipótesis de Inducción:

Supongamos que $L_1 \cup L_2 ... \cup L_n$ es regular y $\exists r$ regex tal que $L_1 \cup L_2 \cup ... \cup L_n = [[r]]$

Paso Inductivo:

Por Demostrar que $L_1 \cup L_2 \cup ... \cup L_n \cup L_{n+1}$ es regular.

$\exists r$ regex tal que $L_{n+1} = [[r`]]$

$$\begin{aligned} L_1 \cup L_2 \cup ... \cup L_n \cup L_{n+1} &= (L_1 \cup L_2 \cup ... \cup L_n) \cup L_{n+1} \\ &= [[r]] \cup [[r`]] \quad \text{Por Hipótesis de Inducción} \\ &= [[r + r`]] \end{aligned}$$

Por definicion tenemos que $r + r`$ es una regex y $[[r + r`]]$ es un lenguaje regular

$\therefore L_1 \cup L_2 \cup ... \cup L_n \cup L_{n+1}$ es regular.

Por lo tanto los lenguajes regulares son cerrados bajo la unión.

6 Diseñar un autómata finito determinista que acepte el lenguaje de las cadenas binarias con un número par de 1's.

(a) Describirlo formalmente

Considérese el autómata $A = \{Q, \Sigma, \delta, q_0, F\}$, donde: $\Sigma = \{0, 1\}, Q = \{q_0, q_1\}, q_0$ el estado inicial, $F = \{q_1\}$ y las transiciones dadas por:

$$\begin{aligned} \delta(q_0, 0) &= q_0 \\ \delta(q_0, 1) &= q_1 \\ \delta(q_1, 0) &= q_1 \\ \delta(q_1, 1) &= q_0 \end{aligned}$$

(b) Dibujar el diagrama de transiciones y la tabla de estados.

Diagrama de transiciones

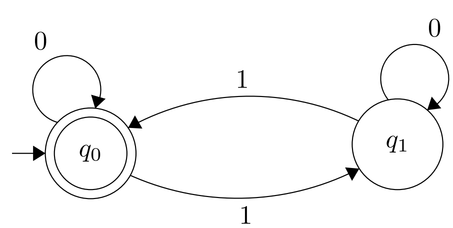


Tabla de estados

f	**0**	**1**
→ q_0	q_0	q_1
q_1	q_1	q_0

(c) Demostrar que el autómata en efecto acepta el lenguaje de cadenas con número par de 1's.

$\rightarrow$

P.D Si A genera a la cadena w, entonces w tiene un numero par de 1's

Haremos inducción sobre el numero de pasos del autómata

Paso base: En 0 pasos, se aplica la transición $\delta(q_0, \epsilon) = q_0$ (por definicion de la funcion de transición), por lo que $w = \epsilon$

Como ϵ tiene 0 1's y es de la forma $0^*(10^*10^*)^*$, entonces $w \in L$

Hipótesis de Inducción: Supongamos que A acepta a x y a y en n y m pasos respectivamente y que $\delta(q_0, x) = \delta(q_0, y) = q_0$ y x y tienen un numero par de 1's.

Paso Inductivo: P.D que se cumple para $w = x\ y$ en $n+m$ pasos y que w tiene un numero par de 1's. Sea $w = x\ y$ tal que $\delta(q_0, x) = \delta(q_0, y) = q_0$, por **H.I**, acepta a x en n pasos y a y en m pasos, por lo que A acepta a w en $n+m$ pasos.

Además, como x y tienen un numero par de 1's, entonces w tiene un numero par de 1's.

$\therefore w \in L$

$\leftarrow$

P.D Si w tiene un numero par de 1's, entonces w es aceptada por A.

Haremos inducción sobre la cadena w

Paso base: Sea $w = \epsilon$ entonces w tiene un numero par de 1's (0), entonces por definicion de transición, $\delta(q_0, 0) = q_0$, que es el estado final.

Por lo tanto ϵ es aceptado por A

Sea $w = 0$ entonces w tiene un numero par de 1's (0), a lo que tenemos que $\delta(q_0, 0^*) = \delta(q_0, 0) = q_0$, que es el estado final.

$\therefore w$ es aceptado por A

H.I Supongamos que x tiene un numero par de 1's, $x \in L$ y $\delta(q_0, x) = q_0$

Paso inductivo: P.D que se cumple para $w = x \cdot a$, donde x tiene un numero par de 1's y $a \in \Sigma$

- Si $a = 0$, entonces por **H.I** y por la funcion de transición de A:
  $\delta(q_0, w) = \delta(q_0, x0) = \delta(\delta(q_0, x), 0) = \delta(q_0, 0) = q_0$
  $\therefore w$ es aceptada por A
- Si $a = 1$, entonces por **H.I** y por la funcion de transición de A:
  $\delta(q_0, w) = \delta(q_0, x1) = \delta(\delta(q_0, x), 1) = \delta(q_0, 1) = q$
  Por lo que w no es aceptada por A, notemos que w tiene un numero impar de 1's.
  Sin embargo, hagamos $u = wb$, con $b \in \Sigma$
  - Si $b = 0$, entonces w sigue teniendo un numero impar de 1's y $\delta(q_0, u) = \delta(q_0, w0) = \delta(\delta(q_0, w), 0) = \delta(q_1, 0) = q_1$
    $\therefore u$ no es aceptada por A
  - Si $b = 1$, entonces w tiene un numero par de 1's y
    $\delta(q_0, u) = \delta(q_0, w1) = \delta(\delta(q_0, w), 1) = \delta(q_1, 1) = q_0$
    $\therefore w$ es aceptada por A

7 Dadas las siguientes expresiones regulares:

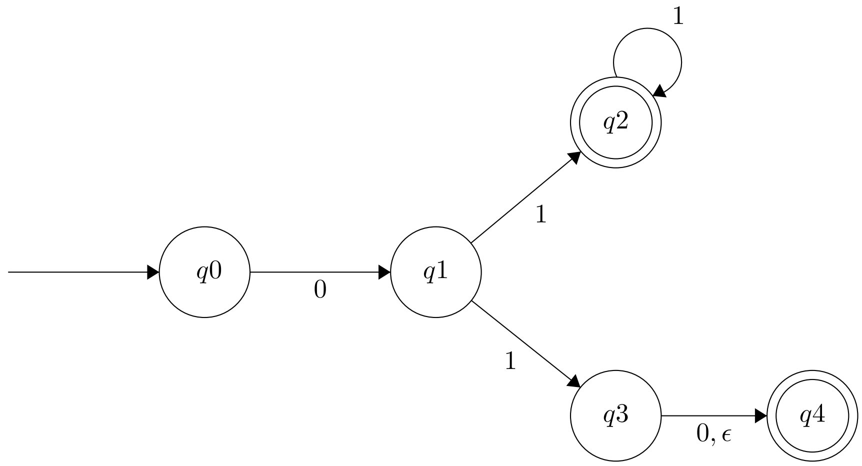
$$R = 01^+ + 010?$$

$$S = 01(1^* + (0 + \epsilon))$$

(a) Mostrar que son equivalentes, desarrollando las expresiones (señalar las propiedades usadas).

$$\begin{aligned}
R &= 01^+ + 010? \\
&= 0(1 \cdot 1^*) + 01(0 + \epsilon) && \text{definicion} \\
&= (01)1^* + 01(0 + \epsilon) && \text{asociatividad de la concatenación} \\
&= 01(1^* + (0 + \epsilon)) && \text{distributividad} \\
&= S
\end{aligned}$$

(b) Describir un autómata finito no determinista con transiciones ϵ que acepte el lenguaje. Dibujar el diagrama de transición y la tabla de estados.
El siguiente autómata es un autómata no determinista con transiciones ϵ que acepta el lenguaje, sin embargo, podemos quitar la transición ϵ y nos quedaría un autómata finito no determinista que sigue aceptando el lenguaje.

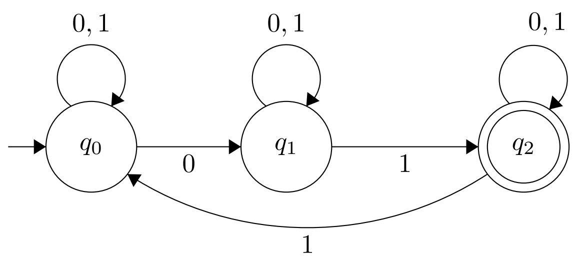


Sea $A = (Q, \Sigma, \delta, q0, F)$ definida como $\Sigma = \{0, 1\},\ Q = \{q0, q1, q2, q3, q4\},\ F = \{q2, q4\}$
Y sean las transiciones dadas por:

	ϵ	**0**	**1**
$\to q_0$	$\varnothing$	$\{q_1\}$	$\varnothing$
q_1	$\varnothing$	$\varnothing$	$\{q_2, q_3\}$
$*q_2$	$\varnothing$	$\varnothing$	$\{q_2\}$
q_3	$\{q_4\}$	$\{q_4\}$	$\varnothing$
$*q_4$	$\varnothing$	$\varnothing$	$\varnothing$

$\delta(q_0, 0) = \{q_1\}$
$\delta(q_1, 1) = \{q_2, q_3\}$
$\delta(q_2, 1) = \{q_2\}$
$\delta(q_3, 0) = \{q_4\}$
$\delta(q_3, \epsilon) = \{q_4\}$

8 A partir del siguiente diagrama de transiciones siguiente del autómata no determinista:



(a) Describirlo formalmente y elaborar su tabla de estados
Sea el AFN $A = (Q, \Sigma, \delta, q_0, F)$ determinado por:

- $Q = \{q_0, q_1, q_2\}$
- $\Sigma = \{0, 1\}$
- q_0 es el estado inicial
- $F = \{q_2\}$
- $\delta : Q \times P(Q)$ dada por:

	0	**1**
$\to q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_1\}$	$\{q_1, q_2\}$
$*q_2$	$\{q_2\}$	$\{q_2, q_0\}$

$\delta(q_0, 0) = \{q_0\}$
$\delta(q_0, 1) = \{q_0, q_2\}$
$\delta(q_1, 0) = \{q_1\}$
$\delta(q_1, 1) = \{q_1, q_2\}$
$\delta(q_2, 0) = \{q_2\}$
$\delta(q_2, 1) = \{q_2, q_0\}$

(b) Convertirlo en un autómata determinista usando el algoritmo de conversión y explicando cada uno de los pasos. Obtener tanto la tabla de estados como el diagrama de transición.
Sea A el AFN anterior, entonces

i. Obtenemos los posibles estados del AFD.

$$Q_D = \{\varnothing, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$$

$$\{q_0\} \text{ es el estado inicial}$$

$$F_D = \{\{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$$

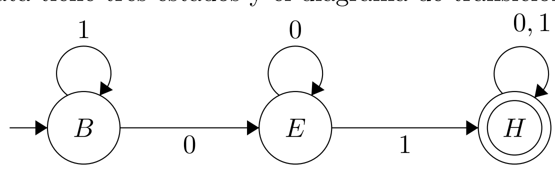
ii. La tabla de transición para δ_D será construida como

	0	**1**
$\varnothing$	$\varnothing$	$\varnothing$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2\}$
$*\{q_2\}$	$\{q_2\}$	$\{q_0, q_2\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$
$*\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$
$*\{q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$

iii. Renombramos los estados y vemos cuáles son alcanzables

	0	**1**
A	A	A
$\rightarrow B$	E	B
C	C	G
*D	D	F
E	E	H
*F	H	F
*G	G	H
*H	H	H

iv. Entonces nuestro autómata tiene tres estados y el diagrama de transiciones es:



(c) Escribir la expresión regular que define este autómata y encontrar una expresión equivalente que sólo use los operadores de concatenación y estrella de Kleene.

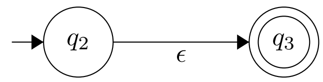
Podemos ver que las cadenas que acepta este autómata son aquellas que tienen al menos un 0 y un 1 y la expresión regular que define este autómata es $(0+1)^*0(0+1)^*1(0+1)^*$.
Usando la propiedad de la cerradura de Kleene que nos dice que $(R+S)^* = (R^*S^*)^*$, la expresión regular $(0+1)^*0(0+1)^*1(0+1)^*$ queda como $(0^*1^*)^*0(0^*1^*)^*1(0^*1^*)^*$, pero como se mencionó anteriormente, estas son las cadenas que tienen al menos un 1 y un 0, por lo que podemos ver a la expresión regular como $1^*0^+1(0+1)^*$, que usando solamente estrella de Kleene y concatenación nos queda $1^*00^*1(0^*1^*)^*$.

9 A partir de la siguiente expresión regular:

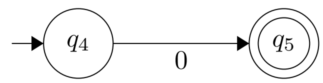
$$(0? + 1)1^+$$

(a) Generar el autómata finito mediante el algoritmo de construcción de Thompson y dibujar el diagrama.
Construimos los pasos base

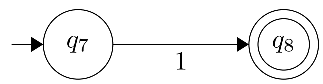
- $\delta(q_2, \epsilon) = q_3$



- $\delta(q_4, 0) = q_5$

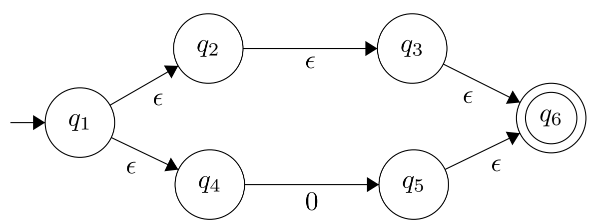


- $\delta(q_7, 1) = q_8$



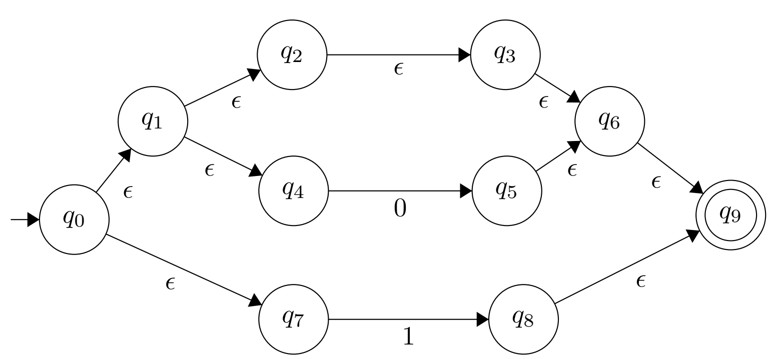
Construimos la expresión para $0? = (0 + \epsilon)$

$$\delta(q_1, \epsilon) = q_2$$
$$\delta(q_1, \epsilon) = q_4$$
$$\delta(q_3, \epsilon) = q_6$$
$$\delta(q_5, \epsilon) = q_6$$



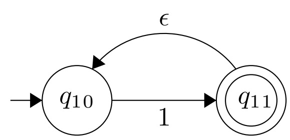
Construimos para $0? + 1$

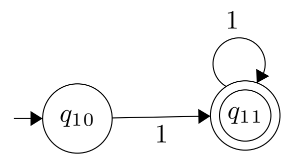
$$\delta(q_0, \epsilon) = q_1$$
$$\delta(q_0, \epsilon) = q_7$$
$$\delta(q_6, \epsilon) = q_9$$
$$\delta(q_8, \epsilon) = q_9$$



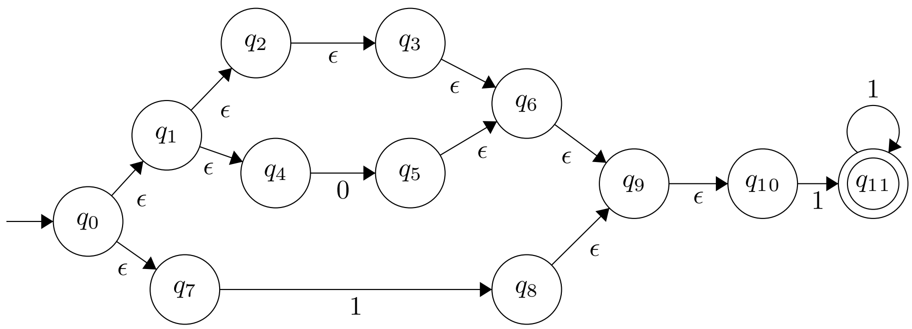
Costruimos para 1^+

$$\delta(q_10, 1) = q_11$$
$$\delta(q_{11}, \epsilon) = q_10 \to \delta(q_{11}, 1) = q_{11}$$





Construimos para toda la expresión



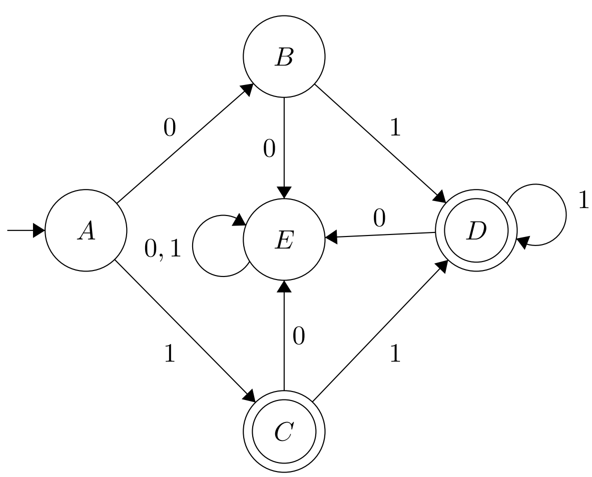
(b) Usar el algoritmo para transformarlo en un AFD. Dibujar el diagrama.
Tenemos que

EClose(q_0) = $\{q_0, q_1, q_2, q_3, q_4, q_6, q_7, q_9, q_{10}\}$
EClose(q_1) = $\{q_1, q_2, q_3, q_4, q_6, q_9, q_{10}\}$
EClose(q_2) = $\{q_2, q_3, q_6, q_9, q_{10}\}$
EClose(q_3) = $\{q_3, q_6, q_9, q_{10}\}$
EClose(q_4) = $\{q_4\}$
EClose(q_5) = $\{q_5, q_6, q_9, q_{10}\}$
EClose(q_6) = $\{q_6, q_9, q_{10}\}$
EClose(q_7) = $\{q_7\}$
EClose(q_8) = $\{q_9, q_{10}\}$
EClose(q_9) = $\{q_9, q_{10}\}$
EClose(q_{10}) = $\{q_{10}\}$
EClose(q_{11}) = $\{q_{11}\}$

Entonces, veremos a que estados nos manda

	0	1
$\to \{q_0, q_1, q_2, q_3, q_4, q_6, q_7, q_9, q_{10}\}$	$\{q_5, q_6, q_9, q_{10}\}$	$\{q_8, q_9, q_{10}, q_{11}\}$
$\{q_5, q_6, q_9, q_{10}\}$	$\emptyset$	$\{q_{11}\}$
$*\{q_8, q_9, q_{10}, q_{11}\}$	$\emptyset$	$\{q_{11}\}$
$*\{q_{11}\}$	$\emptyset$	$\{q_{11}\}$
$\emptyset$	$\emptyset$	$\emptyset$

Ahora haremos el autómata en donde $A = \{q_0, q_1, q_2, q_3, q_4, q_6, q_7, q_9, q_{10}\}, B = \{q_5, q_6, q_9, q_{10}\}, C = \{q_8, q_9, q_{10}, q_{11}\}, D = \{q_{11}\}$ y $E = \emptyset$



Cuya tabla de estados es

	0	1
$\rightarrow A$	B	C
B	E	D
$*C$	E	D
$*D$	E	D
E	E	E

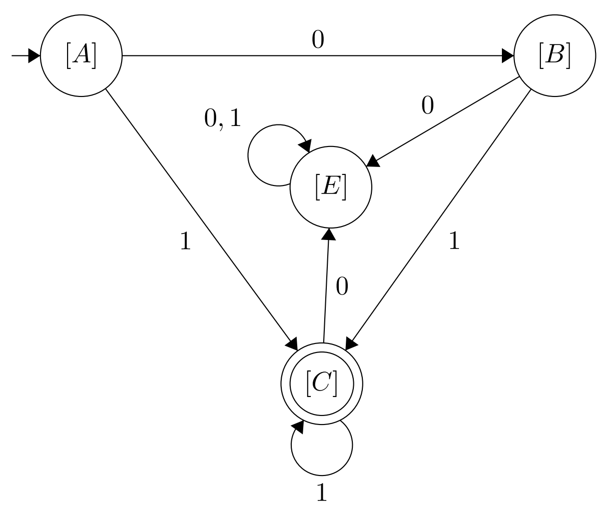
(c) Aplicar el algoritmo de llenado de tabla para obtener estados distinguibles y equivalentes.

B	X			
C	X	X		
D	X	X		
E	X	X	X	X
	A	B	C	D

Donde podemos ver que D y C son equivalentes, así que $[A] = \{A\}, [B] = \{B\}, [C] = \{C, D\}$ y $[E] = \{E\}$

(d) Aplicar el algoritmo para encontrar el autómata mínimo. Dibujar el diagrama.
Tenemos que $Q/ = \{[A], [B], [C], [D]\}$

	0	1
$\rightarrow [A]$	$[B]$	$[C]$
$[B]$	$[E]$	$[C]$
$*[C]$	$[E]$	$[C]$
$[E]$	$[E]$	$[E]$



10 Usando el lema del bombeo, probar que los siguientes lenguajes no son regulares:

(a) $L = \{0^n 1^m : n, m \in \mathbb{N}\}$
Supongamos que L es regular, entonces L cumple el lema del bombeo.

I. Sea $m \in \mathbb{N}$ la constante del lema del bombeo.

II. Construimos $w = 0^0 1^m 2^m = 1^m 2^m$. Claramente $w \in L$.
Como $|xy| \leq m$, entonces xy consiste únicamente de 1's. Como $y \neq \epsilon$, entonces tenemos que $y = 1^a$, con $0 < a \leq m$.
Por el lema del bombeo, la cadena $c = cy^0 z = 1^{m-a} 2^m \in L$, pero $m - a < m$, por lo que $c \notin L$ y no se cumple el lema del bombeo.
Por lo tanto, L no es regular.

(b) $L = \{a^m b^n : 1 \leq m \leq n\}$

Supongamos que L es regular, entonces L cumple el lema del bombeo.

I. Sea $m \geq 1$ la constante del lema del bombeo.

II. Construimos $w = a^m b^m$. Claramente $w \in L$.
Como $|xy| \leq m$, entonces xy consiste únicamente de símbolos a. Por lo tanto, $y = a^u$, con $0 < u \leq m$. Por el lema del bombeo, la cadena $c = cy^{2u}z = a^{m-u}a^{2u}b^m = a^{m+u}b^m \in L$, pero $m + u > m$, por lo que $c \notin L$ y no se cumple el lema del bombeo.
Por lo tanto, L no es regular.

(c) $L = \{a^p : p \text{ es primo}\}$ y $\Sigma = \{a\}$

Supongamos que L es regular, entonces L cumple el lema del bombeo.

I. Sea p primo la constante del lema del bombeo.

II. Construimos $w = a^p$. Como $|xy| \leq p$, entonces $y = a^b$, $0 < b \leq p$. Además $w = a^{b+q}$, con $b + q = p$.

- Si $q = 0$, entonces por el lema del bombeo, $c = a^0 \in L$, pero 0 no es primo, por lo que $c \in L$.
- Si $q = 1$, entonces por el lema del bombeo, $c = a^{3q}a^q = a^{4q}$, pero $4q$ no es primo, por lo que $c \notin L$.
- Si $q \geq 2$, entonces por el lema del bombeo, $c = a^{qb}a^q = a^{q(b+1)} \in L$, pero $q(b + 1)$ no es primo, pues es la multiplicación de dos números, por lo tanto, $c \notin L$.

Como el lema del bombeo se contradice, entonces L no es regular.