

Modelos de Computación



Los Del DGIIM, losdelldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

Modelos de Computación

Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

Granada, 2024-2025

Índice general

1. Relaciones de Problemas	5
1.1. Propiedades de los Lenguajes Regulares	6
1.1.1. Preguntas Tipo Test	58

1. Relaciones de Problemas

1.1. Propiedades de los Lenguajes Regulares

Ejercicio 1.1.1. Determinar si los siguientes lenguajes son regulares o libres de contexto. Justificar las respuestas.

$$1. L_1 = \{0^i b^j \mid i = 2j \text{ ó } 2i = j\}$$

Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^{2n} b^n \in L_1$ con $|z| = 3n \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, b\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{2n-k-l} b^n \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+2n-k-l} b^n = 0^{2n+l} b^n \notin L_1$, ya que, como $l \geq 1$:

$$2n + l \neq 2n \quad \text{y} \quad 2(2n + l) \neq n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Veamos ahora que sí es libre de contexto. Consideramos la gramática $G = (\{S, S_1, S_2\}, \{0, b\}, P, S)$ con P definido por:

$$\begin{aligned} S &\rightarrow S_1 \mid S_2 \\ S_1 &\rightarrow 00S_1 b \mid \varepsilon \\ S_2 &\rightarrow 0S_2 b b \mid \varepsilon \end{aligned}$$

Tenemos que G es una gramática libre de contexto tal que $\mathcal{L}(G) = L_1$, por lo que L_1 es libre de contexto.

$$2. L_2 = \{uu^{-1} \mid u \in \{0, 1\}^*, |u| \leq 1000\}$$

Consideramos el lenguaje auxiliar:

$$L'_2 = \{u \in \{0, 1\}^* \mid |u| \leq 2 \cdot 1000\}$$

Veamos que L'_2 es finito. Como el número de combinaciones de n elementos de $\{0, 1\}$ es 2^n , entonces el número de palabras de longitud menor o igual a $2 \cdot 1000$ es:

$$|L'_2| = \sum_{i=0}^{2 \cdot 1000} 2^i < \infty$$

Por tanto, como $L_2 \subset L'_2$ finito, tenemos que L_2 es finito y por tanto regular.

$$3. L_3 = \{uu^{-1} \mid u \in \{0, 1\}^*, |u| \geq 1000\}$$

Sabemos que el siguiente lenguaje es independiente del contexto:

$$L'_3 = \{uu^{-1} \mid u \in \{0, 1\}^*\}$$

Además, tenemos que $L'_3 = L_2 \cup L_3$. Supongamos que L_3 es regular. Entonces, como L_2 es regular, tendríamos que L'_3 es regular, lo cual es una contradicción.

Por tanto, L_3 no es regular. Para ver que es libre de contexto, consideramos la gramática $G = (V, \{0, 1\}, P, S)$ con:

$$V = \{S\} \cup \{A_i \mid i \in \{1, \dots, 1000\}\}$$

Tenemos que P está definido por:

$$\begin{aligned} S &\rightarrow 0A_10 \mid 1A_11, \\ A_i &\rightarrow 0A_{i+1}0 \mid 1A_{i+1}, \quad i \in \{1, \dots, 999\}, \\ A_{1000} &\rightarrow 0A_{1000}0 \mid 1A_{1000}1 \mid \varepsilon \end{aligned}$$

Notemos que, en A_i , ya hemos leído i caracteres de u (la palabra que forma la mitad del palíndromo). Una vez hemos llegado a A_{1000} , hemos leído 1000 caracteres de u . Por tanto, podemos añadir los que queramos sin restricción, y podemos también terminar. Como $L_3 = \mathcal{L}(G)$, tenemos que L_3 es independiente del contexto.

4. $L_4 = \{0^i 1^j 2^k \mid i = j \text{ ó } j = k\}$

Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^n 1^n 2^{2n} \in L_4$ con $|z| = 3n \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1, 2\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{n-k-l} 1^n 2^{2n} \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+n-k-l} 1^n 2^{2n} = 0^{n+l} 1^n 2^{2n} \notin L_4$, ya que, como $l \geq 1$:

$$n + l \neq n \quad \text{y} \quad n \neq 2n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Veamos ahora que sí es libre de contexto. Consideramos la gramática $G = (V, \{0, 1, 2\}, P, S)$ donde $V = \{S, S_1, S_2, A_0, A_2\}$ y P está definido por:

$$\begin{aligned} S &\rightarrow S_1 A_2 \mid A_0 S_2 \\ S_1 &\rightarrow 0 S_1 1 \mid \varepsilon \\ A_2 &\rightarrow 2 A_2 \mid \varepsilon \\ S_2 &\rightarrow 1 S_2 2 \mid \varepsilon \\ A_0 &\rightarrow 0 A_0 \mid \varepsilon \end{aligned}$$

Tenemos que G es una gramática libre de contexto tal que $\mathcal{L}(G) = L_4$, por lo que L_4 es libre de contexto.

Ejercicio 1.1.2. Determinar qué lenguajes son regulares o libres de contexto de los siguientes:

1. $\{u0u^{-1} \mid u \in \{0, 1\}^*\}$

Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^n 10^n \in L$ con $|z| = 2n + 1 \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{n-k-l} 10^n \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+n-k-l}10^n = 0^{n+l}10^n \notin L$, ya que, como $l \geq 1$:

$$n + l \neq n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Veamos ahora que sí es libre de contexto. Consideramos la gramática $G = (\{S\}, \{0, 1\}, P, S)$, con P definido por:

$$S \rightarrow 0S01S1 \mid 0 \mid 1$$

Tenemos que G es una gramática libre de contexto tal que $\mathcal{L}(G) = L$, por lo que L es libre de contexto.

2. Números en binario que sean múltiplos de 4

Tenemos que todos los números en binario que son múltiplos de 4 terminan en 00, por lo que vienen dados por la siguiente expresión regular:

$$0^* + (1 + 0)^*00$$

Notemos que hemos incluido 0^* , porque el 0 también es múltiplo de 4. Por tanto, es regular.

3. Palabras de $\{0, 1\}^*$ que no contienen la subcadena 0110.

Notemos que podríamos dar un autómata que reconociese ese lenguaje, pero no es la opción más sencilla. Veamos en primer lugar que el lenguaje formado por las palabras que sí contienen la subcadena 0110 es regular dando una expresión regular asociada a él:

$$(0 + 1)^*0110(0 + 1)^*$$

Como el lenguaje descrito es su complementario y el complementario de un regular es regular, tenemos que el lenguaje dado es regular.

Ejercicio 1.1.3. Determinar qué lenguajes son regulares y qué lenguajes son libres de contexto entre los siguientes:

1. Conjunto de palabras sobre el alfabeto $\{0, 1\}$ en las que cada 1 va precedido por un número par de ceros.

Un reconocedor del lenguaje es el autómata de la Figura 1.1, que tiene los siguientes estados:

- q_0 : Llevo un número par de ceros consecutivos, puedo leer un 1.
- q_1 : Llevo un número impar de ceros consecutivos, no puedo leer un 1.
- q_2 : Acabo de leer un 1.

Por tanto, como hemos dado un autómata que reconoce el lenguaje, es regular.

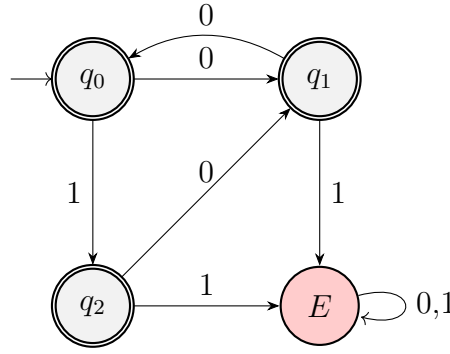


Figura 1.1: Autómata que reconoce el lenguaje del ejercicio 1.1.3.1.

2. Conjunto $\{0^i 1^{2j} 0^{i+j} \mid i, j \geq 0\}$

Usaremos el Lema de Bombeo para demostrar que no es regular. Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^n 1^{2n} 0^{2n}$ con $|z| = 5n \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{n-k-l} 1^{2n} 0^{2n} \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+n-k-l} 1^{2n} 0^{2n} = 0^{n+l} 1^{2n} 0^{2n} \notin L$, ya que, como $l \geq 1$:

$$2n \neq n + n + l$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Veamos ahora que es libre de contexto. Consideramos la gramática $G = (\{S, X\}, \{0, 1\}, P, S)$, con P definido por:

$$\begin{aligned} S &\rightarrow 11S0 \mid X, \\ X &\rightarrow 0X0 \mid \varepsilon. \end{aligned}$$

Tenemos que G es una gramática libre de contexto tal que $\mathcal{L}(G) = L$, por lo que L es libre de contexto.

3. Conjunto $\{0^i 1^j 0^{i+j} \mid i, j \geq 0\}$

Usaremos el Lema de Bombeo para demostrar que no es regular. Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^n 1^n 0^{n^2}$ con $|z| = n + n + n^2 \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{n-k-l} 1^n 0^{n^2} \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+n-k-l} 1^n 0^{n^2} = 0^{n+l} 1^n 0^{n^2} \notin L$, ya que, como $l \geq 1$:

$$(n + l) \cdot n = n^2 + nl \neq n^2$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Además, este lenguaje no es libre de contexto (algo que aún no podemos demostrar).

Ejercicio 1.1.4. Determina si los siguientes lenguajes son regulares. Encuentra una gramática que los genere o un reconocedor que los acepte.

1. $L_1 = \{0^i 1^j \mid j < i\}$.

Usaremos el Lema de Bombeo para demostrar que no es regular. Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^{n+1} 1^n \in L_1$ con $|z| = 2n + 1 \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{n+1-k-l} 1^n \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 0$, tenemos que $uv^i w = 0^{k+n+1-k-l} 1^n = 0^{n+1-l} 1^n \notin L_1$, ya que:

$$n < n + 1 - l \iff l < 1$$

Pero esto es una contradicción, ya que $l \geq 1$. Por tanto, por el recíproco del Lema de Bombeo, no es regular. Veamos ahora que es libre de contexto. Consideramos la gramática $G = (\{S\}, \{0, 1\}, P, S)$, con P definido por:

$$\begin{aligned} S &\rightarrow 0S \mid 0S' \\ S' &\rightarrow 0S'1 \mid \varepsilon \end{aligned}$$

Tenemos que G es una gramática libre de contexto tal que $\mathcal{L}(G) = L_1$, por lo que L_1 es libre de contexto. Notemos que la producción $S \rightarrow 0S'$ fuerza a que haya al menos un 0 más que 1, y la producción $S' \rightarrow 0S'1$ permite que la diferencia no sea de una sola unidad, sino que pueda ser mayor.

2. $L_2 = \{001^i 0^j \mid i, j \geq 1\}$.

Tenemos que un reconocedor de L_2 es:

$$001^+ 0^+$$

Por tanto, tenemos que L_2 es regular.

3. $L_3 = \{010u \mid u \in \{0, 1\}^*, u \text{ no contiene la subcadena } 010\}$.

Sea $L' = \{u \in \{0, 1\}^* \mid u \text{ contiene la subcadena } 010\}$. Entonces sabemos que L' es regular con reconocedor:

$$(0 + 1)^* \mathbf{010} (0 + 1)^*$$

Por tanto, $\overline{L'}$ es regular, por lo que está asociado a una expresión regular, sea esta \tilde{r} . Entonces, L_3 es regular y está asociado a la expresión regular:

$$010\tilde{r}$$

Ejercicio 1.1.5. Sea el alfabeto $A = \{0, 1, +, =\}$, demostrar que el lenguaje

$$\text{ADD} = \{x = y + z \mid x, y, z \text{ son números en binario, y } x \text{ es la suma de } y \text{ y } z\}$$

no es regular.

Usaremos el Lema de Bombeo para demostrar que no es regular. Para todo $n \in \mathbb{N}$, consideramos la palabra $w = [1^n = 0 + 1^n]$, donde hemos empleado los corchetes para facilitar la notación (ya que el $=$ lo estamos usando para igualdades entre cadenas y entre números en binario). Tenemos que $|w| = n + 3 + n \geq n$. Toda descomposición $w = uvw$, con $u, v, w \in \{0, 1, +, =\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 1^k \quad v = 1^l \quad w = [1^{n-k-l} = 0 + 1^n] \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = [1^{k+2l+n-k-l} = 0 + 1^n] = [1^{n+l} = 0 + 1^n] \notin \text{ADD}$, ya que, como $l \geq 1$, en números binarios:

$$1^{n+l} \neq 0 + 1^n = 1^n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular.

Ejercicio 1.1.6. Determinar si los siguientes lenguajes son regulares o no:

1. $L = \{uvu^{-1} \mid u, v \in \{0, 1\}^*\}$.

Veamos en primer lugar que $L = \{0, 1\}^*$ por doble inclusión:

⊂) Se tiene trivialmente que $L \subset \{0, 1\}^*$.

⊃) Sea $w \in \{0, 1\}^*$. Entonces, podemos tomar $u = \varepsilon$ y $v = w$ para obtener $w \in L$.

Por tanto, tenemos que L es regular, con reconocedor:

$$(0 + 1)^*$$

2. L es el lenguaje sobre el alfabeto $\{0, 1\}$ formado de las palabras de la forma $u0v$ donde u^{-1} es un prefijo de v .

Usaremos el Lema de Bombeo para demostrar que no es regular. Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 1^n 0 1^n \in L$ con $|z| = 2n + 1 \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 1^k \quad v = 1^l \quad w = 1^{n-k-l} 0 1^n \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Para $i = 2$, tenemos que $uv^i w = 1^{k+2l+n-k-l} 0 1^n = 1^{n+l} 0 1^n \notin L$, ya que, como $l \geq 1$:

$$n + l \neq n \implies (1^{n+l})^{-1} = 1^{n+l} \neq 1^n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular.

3. L es el lenguaje sobre el alfabeto $\{0, 1\}$ formado por las palabras en las que el tercer símbolo empezando por el final es un 1.

Este lenguaje es regular, con reconocedor:

$$(0 + 1)^* \mathbf{1} (0 + 1) (0 + 1)$$

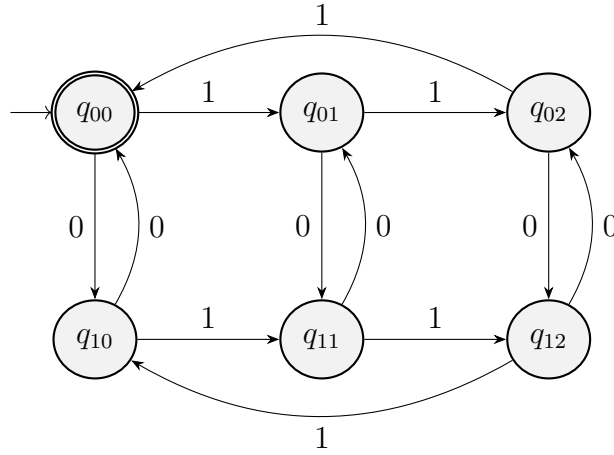


Figura 1.2: Autómata que reconoce el lenguaje del ejercicio 1.1.7.1.

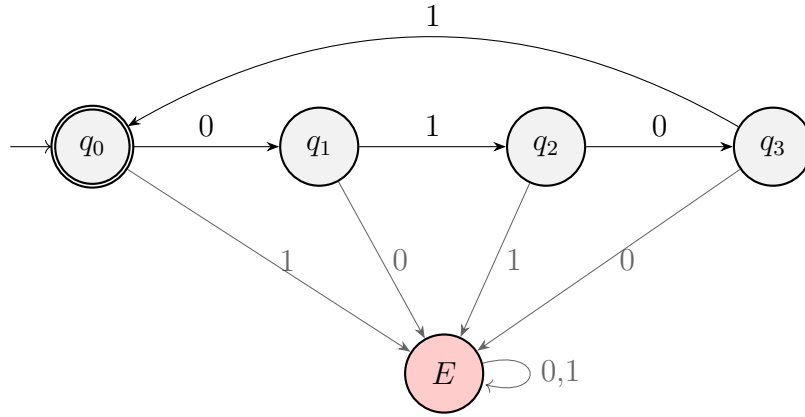


Figura 1.3: Autómata que reconoce el lenguaje del ejercicio 1.1.7.2.

Ejercicio 1.1.7. Obtener autómatas finitos determinísticos para los siguientes lenguajes sobre el alfabeto $\{0, 1\}$.

1. Palabras en las que el número de 1 es múltiplo de 3 y el número de 0 es par.
El estado q_{ij} para $i = 0, 1, 2$ y $j = 0, 1$ indica que:

- $n_1(u) \bmod 3 = i$,
- $n_0(u) \bmod 2 = j$.

Entonces, el autómata es el de la Figura 1.2.

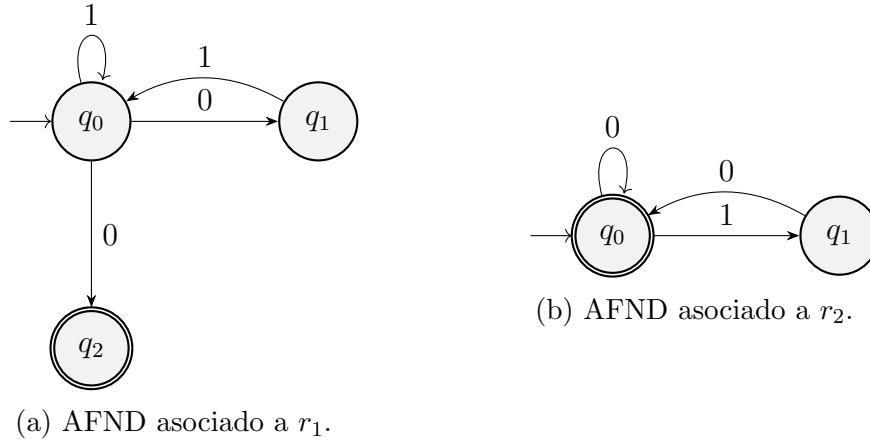
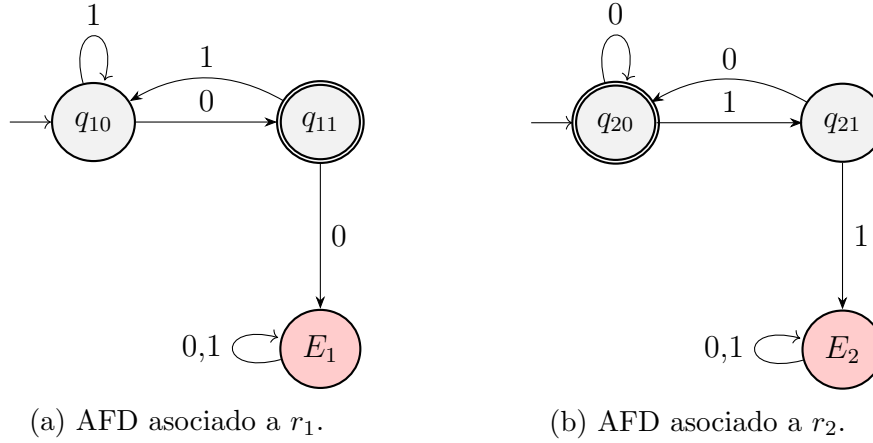
2. $\{(01)^{2i} \mid i \geq 0\}$

El autómata es el de la Figura 1.3.

3. $L_3 = \{(0^{2i}1^{2i}) \mid i \geq 0\}$

Veamos que este lenguaje no es regular con el Lema de Bombeo. Para todo $n \in \mathbb{N}$, consideramos la palabra $z = 0^{2n}1^{2n} \in L_3$ con $|z| = 4n \geq n$. Toda descomposición $z = uvw$, con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe cumplir que:

$$u = 0^k \quad v = 0^l \quad w = 0^{2n-k-l}1^{2n} \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, \quad l \geq 1, \quad k + l \leq n$$

Figura 1.4: AFND asociados a las expresiones regulares r_1 y r_2 .Figura 1.5: AFD asociados a las expresiones regulares r_1 y r_2 .

Para $i = 2$, tenemos que $uv^i w = 0^{k+2l+2n-k-l}1^{2n} = 0^{2n+l}1^{2n} \notin L_3$, ya que, como $l \geq 1$:

$$2n + l \neq 2n$$

Por tanto, por el recíproco del Lema de Bombeo, no es regular. Por tanto, no es posible construir un autómata finito determinístico que reconozca L_3 .

Ejercicio 1.1.8. Dar una expresión regular para la intersección de los lenguajes asociados a las expresiones regulares $(01 + 1)^*0$ y $(10 + 0)^*$. Se valorará que se construya el autómata que acepta la intersección de estos lenguajes, se minimice y, a partir del resultado, se construya la expresión regular.

Sea $r_1 = (01 + 1)^*0$ y $r_2 = (10 + 0)^*$. Construimos los AFND asociados a r_1 y r_2 , mostrados en las figuras 1.4a y 1.4b, respectivamente.

Para poder aplicar el algoritmo de intersección de autómatas, antes hemos de convertir los autómatas en AFD. Los AFD asociados a r_1 y r_2 son los de las figuras 1.5a y 1.5b, respectivamente.

Por tanto, el AFD que acepta la intersección de los lenguajes asociados a r_1 y r_2 es el de la Figura 1.6.

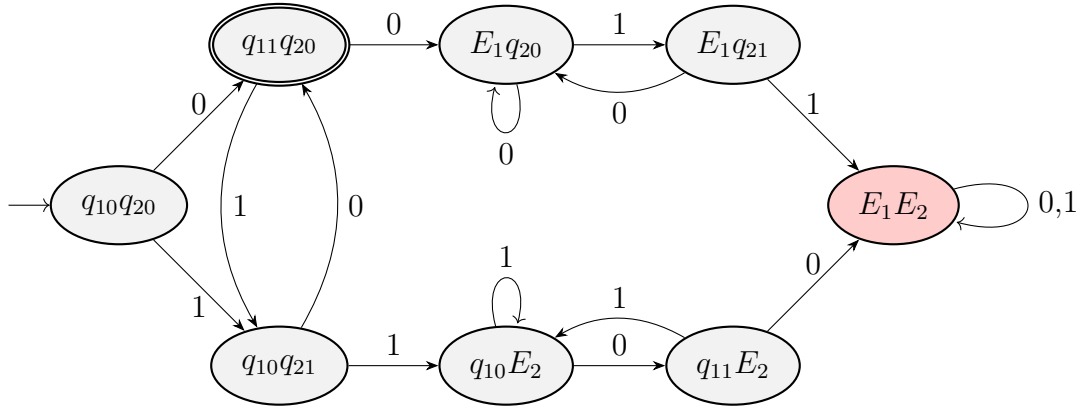


Figura 1.6: AFD que acepta la intersección de los lenguajes asociados a r_1 y r_2 .

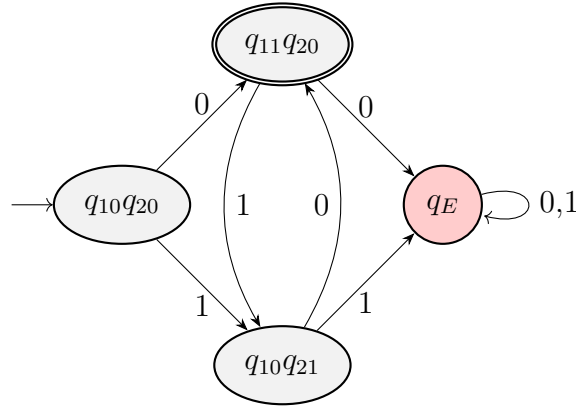


Figura 1.7: AFD minimal que acepta la intersección de los lenguajes asociados a r_1 y r_2 .

Para minimizarlo, consideramos en primer lugar que los siguientes estados son indistinguibles:

$$q_E := \{E_1q_{20}, E_1q_{21}, q_{10}E_2, q_{11}E_2, E_1E_2\}$$

Estos son indistinguibles puesto que desde ninguno de ellos se puede llegar a un estado final. Por tanto, el AFD minimal es el de la Figura 1.7.

Tenemos que es minimal, puesto que todos los estados son alcanzables y no hay estados distinguibles:

- $q_{11}q_{20}$ es distinguible del resto de estados por ser el único estado final.
- q_E es distinguible de $q_{10}q_{20}$ y $q_{10}q_{21}$, ya que leyendo un 0:

$$\delta(q_E, 0) = q_E \notin F \quad \delta(q_{10}q_{20}, 0) = \delta(q_{10}q_{21}, 0) = q_{11}q_{20} \in F$$

- $q_{10}q_{20}$ y $q_{10}q_{21}$ son distinguibles, ya que leyendo 10:

$$\delta(q_{10}q_{20}, 1) = q_{11}q_{20} \in F \quad \delta(q_{10}q_{21}, 1) = q_E \notin F$$

Por tanto, el AFD minimal que acepta la intersección de los lenguajes asociados a r_1 y r_2 es el de la Figura 1.7. Para obtener la expresión regular asociada, resolvemos

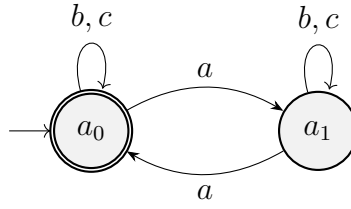


Figura 1.8: AFD que acepta palabras con número par de a 's.

el sistema de ecuaciones:

$$\begin{aligned} q_{10}q_{20} &= 0q_{11}q_{20} + 1q_{10}q_{21} \\ q_{11}q_{20} &= 0q_E + 1q_{10}q_{21} + \varepsilon \\ q_{10}q_{21} &= 0q_{11}q_{20} + 1q_E \\ q_E &= 0q_E + 1q_E = (0 + 1)q_E \end{aligned}$$

Por el Lema de Arden, como $q_E = (0 + 1)q_E + \emptyset$, tenemos que $q_E = (0 + 1)^*\emptyset = \emptyset$. Sustituyendo en las ecuaciones anteriores, obtenemos:

$$\begin{aligned} q_{10}q_{20} &= 0q_{11}q_{20} + 1q_{10}q_{21} \\ q_{11}q_{20} &= 1q_{10}q_{21} + \varepsilon \\ q_{10}q_{21} &= 0q_{11}q_{20} \end{aligned}$$

Sustituyendo $q_{10}q_{21}$ en la segunda ecuación, obtenemos:

$$q_{11}q_{20} = 1q_{11}q_{20} + \varepsilon \implies q_{11}q_{20} = 10q_{11}q_{20} + \varepsilon = (10)^*\varepsilon = (10)^*$$

Sustituyendo ambos en la primera ecuación, obtenemos:

$$q_{10}q_{20} = 0(10)^* + 10q_{11}q_{20} = 0(10)^* + 10(10)^* = (0 + 10)(10)^*$$

Por tanto, la expresión regular asociada al AFD minimal de la Figura 1.7 es

$$(0 + 10)(10)^*.$$

Ejercicio 1.1.9. Construir un Autómata Finito Determinista Minimal que acepte el lenguaje sobre el alfabeto $\{a, b, c\}$ de todas aquellas palabras que verifiquen simultáneamente las siguientes condiciones.

1. La palabra contiene un número par de a 's.
2. La longitud de la palabra es un múltiplo de 3.
3. La palabra no contiene la subcadena abc .

La condición 1 se puede cumplir con el autómata de la Figura 1.8.

La condición 2 se puede cumplir con el autómata de la Figura 1.9.

La condición 3 se puede cumplir con el autómata de la Figura 1.10.

El autómata producto tiene los siguientes estados:

$$\begin{aligned} Q &= \{a_i b_j c_k \mid i \in \{0, 1\}, j \in \{0, 1, 2\}, k \in \{0, 1, 2, 3\}\} \\ F &= \{a_0 b_0 c_0, a_0 b_0 c_1, a_0 b_0 c_2\} \end{aligned}$$

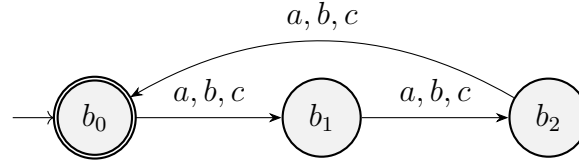
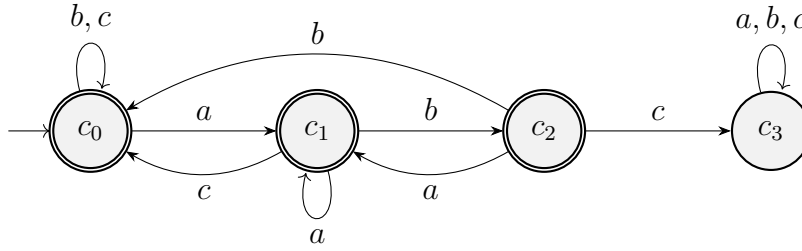


Figura 1.9: AFD que acepta palabras de longitud múltiplo de 3.

Figura 1.10: AFD que acepta palabras que no contienen la subcadena abc .

	1	2	3	4	5	6	7	8	9	10	11	12
δ	$a_0b_0c_0$	$a_0b_0c_1$	$a_0b_0c_2$	$a_0b_0c_3$	$a_0b_1c_0$	$a_0b_1c_1$	$a_0b_1c_2$	$a_0b_1c_3$	$a_0b_2c_0$	$a_0b_2c_1$	$a_0b_2c_2$	$a_0b_2c_3$
a	$a_1b_1c_1$	$a_1b_1c_1$	$a_1b_1c_1$	$a_1b_1c_3$	$a_1b_2c_1$	$a_1b_2c_1$	$a_1b_2c_1$	$a_1b_2c_3$	$a_1b_0c_1$	$a_1b_0c_1$	$a_1b_0c_1$	$a_1b_0c_3$
b	$a_0b_1c_0$	$a_0b_1c_2$	$a_0b_1c_0$	$a_0b_1c_3$	$a_0b_2c_0$	$a_0b_2c_2$	$a_0b_2c_0$	$a_0b_2c_3$	$a_0b_0c_0$	$a_0b_0c_2$	$a_0b_0c_0$	$a_0b_0c_3$
c	$a_0b_1c_0$	$a_0b_1c_0$	$a_0b_1c_3$	$a_0b_1c_3$	$a_0b_2c_0$	$a_0b_2c_0$	$a_0b_2c_3$	$a_0b_2c_3$	$a_0b_0c_0$	$a_0b_0c_0$	$a_0b_0c_3$	$a_0b_0c_3$
	13	14	15	16	17	18	19	20	21	22	23	24
δ	$a_1b_0c_0$	$a_1b_0c_1$	$a_1b_0c_2$	$a_1b_0c_3$	$a_1b_1c_0$	$a_1b_1c_1$	$a_1b_1c_2$	$a_1b_1c_3$	$a_1b_2c_0$	$a_1b_2c_1$	$a_1b_2c_2$	$a_1b_2c_3$
a	$a_0b_1c_1$	$a_0b_1c_1$	$a_0b_1c_1$	$a_0b_1c_3$	$a_0b_2c_1$	$a_0b_2c_1$	$a_0b_2c_1$	$a_0b_2c_3$	$a_0b_0c_1$	$a_0b_0c_1$	$a_0b_0c_1$	$a_0b_0c_3$
b	$a_1b_1c_0$	$a_1b_1c_2$	$a_1b_1c_0$	$a_1b_1c_3$	$a_1b_2c_0$	$a_1b_2c_2$	$a_1b_2c_0$	$a_1b_2c_3$	$a_1b_0c_0$	$a_1b_0c_2$	$a_1b_0c_0$	$a_1b_0c_3$
c	$a_1b_1c_0$	$a_1b_1c_0$	$a_1b_1c_3$	$a_1b_1c_3$	$a_1b_2c_0$	$a_1b_2c_0$	$a_1b_2c_3$	$a_1b_2c_3$	$a_1b_0c_0$	$a_1b_0c_0$	$a_1b_0c_3$	$a_1b_0c_3$

Tabla 1.1: Transiciones del autómata producto para el ejercicio 1.1.9.

2	×																		
3	(9, 10)	×																	
5	×	×	×																
6	×	×	×	×															
7	×	×	×	×	×														
9	×	×	×	×	×	×													
10	×	×	×	×	×	×	×												
11	×	×	×	×	×	×	×	×											
13	×	×	×	×	×	×	×	×	×										
14	×	×	×	×	×	×	×	×	×	×									
15	×	×	×	×	×	×	×	×	×	×	×								
17	×	×	×	×	×	×	×	×	×	×	×	×							
18	×	×	×	×	×	×	×	×	×	×	×	×	×						
19	×	×	×	×	×	×	×	×	×	(21, 22)	×	×	×	×	×				
21	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×			
22	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		
23	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	
c_3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	1	2	3	5	6	7	9	10	11	13	14	15	17	18	19	21	22	23	

El autómata producto es $M = (Q, \{a, b, c\}, \delta, a_0b_0c_0, F)$, donde δ viene dado por la Tabla 1.1. Además, todos sus estados son accesibles.

La minimización del autómata producto se muestra en la Figura 1.2.

Por tanto, el autómata minimal que acepta el lenguaje del ejercicio 1.1.9 es M , pero unificando los estados previamente descritos en c_3 . Debido al gran número de estados (19), el grafo de dicho autómata no se muestra.

$$(a+b)^*(aa+bb)(a+b)^*$$

Convertimos el AFND en un AFD, mostrado en la Figura 1.12.

No obstante, este no es minimal. En primer lugar, vemos que los estados $q_0q_1q_3$ y $q_0q_2q_3$ son indistinguibles, ya que para cualquier palabra $w \in \{a, b\}^*$:

$$\delta(q_0q_1q_3, w) \in F \quad \delta(q_0q_2q_3, w) \in F$$

Por tanto, notemos $q_F = \{q_0q_1q_3, q_0q_2q_3\}$. El AFD minimal es el de la Figura 1.13.

- q_F es distinguible del resto de estados por ser el único estado final.
- q_0q_1 y q_0q_2 son distinguibles, ya que leyendo un a :

$$\delta(q_0q_1, a) = q_F \in F \qquad \delta(q_0q_2, a) = q_0q_1 \notin F$$

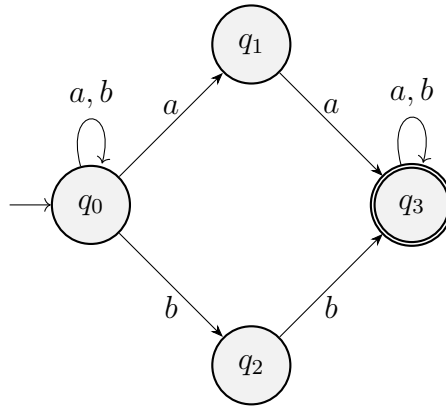


Figura 1.11: AFND asociado a la expresión regular $(a+b)^*(aa+bb)(a+b)^*$.

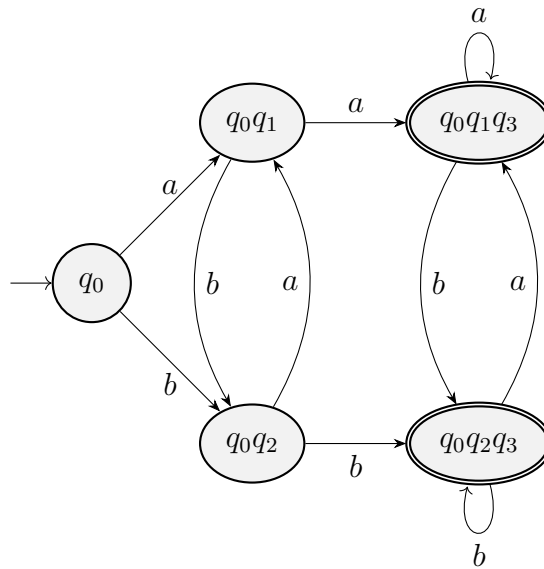


Figura 1.12: AFD asociado a la expresión regular $(a+b)^*(aa+bb)(a+b)^*$.

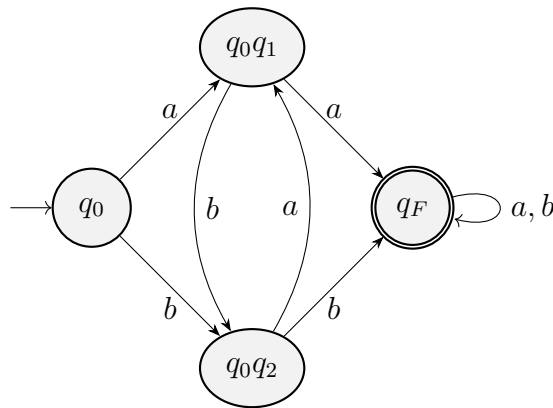


Figura 1.13: AFD minimal asociado a la expresión regular $(a+b)^*(aa+bb)(a+b)^*$.

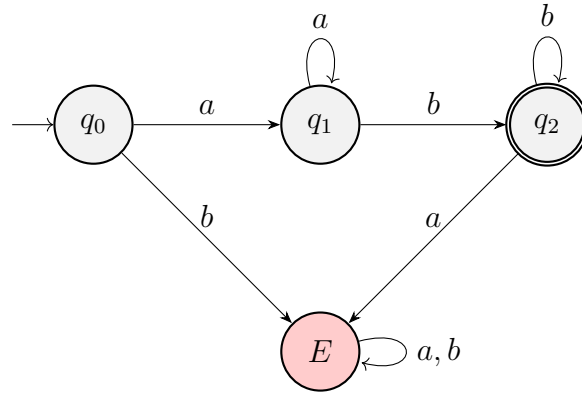


Figura 1.14: AFD asociado al lenguaje a^+b^+ del ejercicio 1.1.11.1.

- q_0 y q_0q_1 son distinguibles, ya que leyendo un a :

$$\delta(q_0, a) = q_0q_1 \notin F \quad \delta(q_0q_1, a) = q_F \in F$$

- q_0 y q_0q_2 son distinguibles, ya que leyendo un b :

$$\delta(q_0, b) = q_0q_2 \notin F \quad \delta(q_0q_2, b) = q_F \in F$$

Por tanto, el AFD minimal es el de la Figura 1.13.

Ejercicio 1.1.11. Para cada uno de los siguientes lenguajes regulares, encontrar el autómata minimal asociado, y a partir de dicho autómata minimal, determinar la gramática regular que genera el lenguaje:

1. a^+b^+

En primer lugar, construimos el AFD asociado al lenguaje, mostrado en la Figura 1.14.

Veamos que este es minimal:

- q_2 es distinguible del resto de estados por ser el único estado final.
- q_0 y q_1 son distinguibles, ya que leyendo un b :

$$\delta(q_0, b) = E \notin F \quad \delta(q_1, b) = q_2 \in F$$

- q_0 y E son distinguibles, ya que leyendo un ab :

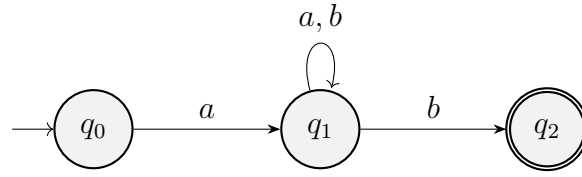
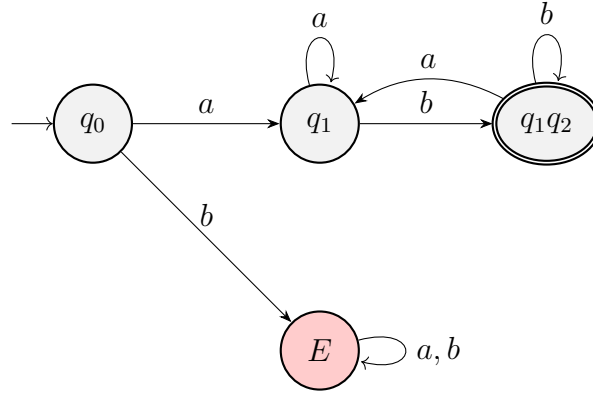
$$\delta^*(q_0, ab) = q_2 \in F \quad \delta^*(E, ab) = E \notin F$$

- q_1 y E son distinguibles, ya que leyendo un b :

$$\delta(q_1, b) = q_2 \in F \quad \delta(E, b) = E \notin F$$

Por tanto, el AFD minimal es el de la Figura 1.14. La gramática regular que genera el lenguaje es $G = (\{q_0, q_1, q_2\}, \{a, b\}, P, \{q_0\})$ con P :

$$\begin{aligned} q_0 &\longrightarrow aq_1 \\ q_1 &\longrightarrow aq_1 \mid bq_2 \\ q_2 &\longrightarrow bq_2 \mid \varepsilon \end{aligned}$$

Figura 1.15: AFND asociado al lenguaje $a(a+b)^*b$ del ejercicio 1.1.11.2.Figura 1.16: AFD asociado al lenguaje $a(a+b)^*b$ del ejercicio 1.1.11.2.2. $a(a+b)^*b$

En primer lugar, construimos el AFND asociado al lenguaje, mostrado en la Figura 1.15.

Convertimos el AFND en un AFD, mostrado en la Figura 1.16.

Veamos que este es minimal:

- q_1q_2 es distinguible del resto de estados por ser el único estado final.
- q_0 y q_1 son distinguibles, ya que leyendo un b :

$$\delta(q_0, b) = E \notin F \quad \delta(q_1, b) = q_1q_2 \in F$$

- q_0 y E son distinguibles, ya que leyendo un ab :

$$\delta^*(q_0, ab) = q_1q_2 \in F \quad \delta^*(E, ab) = E \notin F$$

- q_1 y E son distinguibles, ya que leyendo un b :

$$\delta(q_1, b) = q_1q_2 \in F \quad \delta(E, b) = E \notin F$$

Por tanto, el AFD minimal es el de la Figura 1.16. La gramática regular que genera el lenguaje es $G = (\{q_0, q_1, q_2\}, \{a, b\}, P, \{q_0\})$ con P :

$$\begin{aligned} q_0 &\longrightarrow aq_1 \\ q_1 &\longrightarrow aq_1 \mid bq_2 \\ q_2 &\longrightarrow bq_2 \mid aq_1 \mid \varepsilon \end{aligned}$$

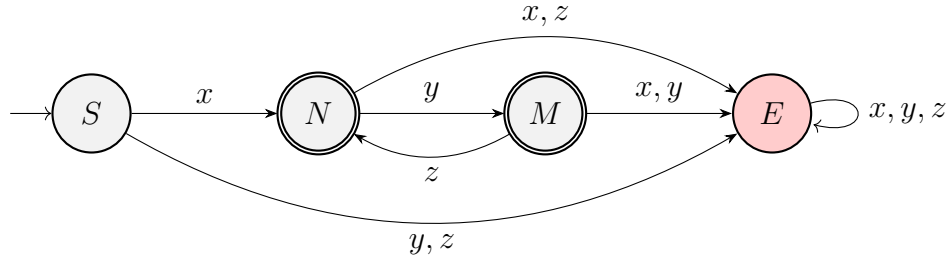


Figura 1.17: AFD asociado al lenguaje $\mathcal{L}(G)$ del ejercicio 1.1.12.

Ejercicio 1.1.12. Considera la gramática cuyas producciones se presentan a continuación y donde el símbolo inicial es S :

$$\begin{aligned} S &\rightarrow xN \mid x \\ N &\rightarrow yM \mid y \\ M &\rightarrow zN \mid z \end{aligned}$$

1. Escribe el diagrama de transiciones para el AFD que acepte el lenguaje $\mathcal{L}(G)$ generado por G .

Las siguientes producciones generan el mismo lenguaje:

$$\begin{aligned} S &\rightarrow xN \\ N &\rightarrow yM \mid \varepsilon \\ M &\rightarrow zN \mid \varepsilon \end{aligned}$$

Por tanto, el AFD asociado al lenguaje $\mathcal{L}(G)$ es el de la Figura 1.17.

2. Encuentra una gramática regular por la izquierda que genere ese mismo lenguaje $\mathcal{L}(G)$.

La expresión regular asociada al lenguaje $\mathcal{L}(G)$ es:

$$x(yz)^*(y + \varepsilon)$$

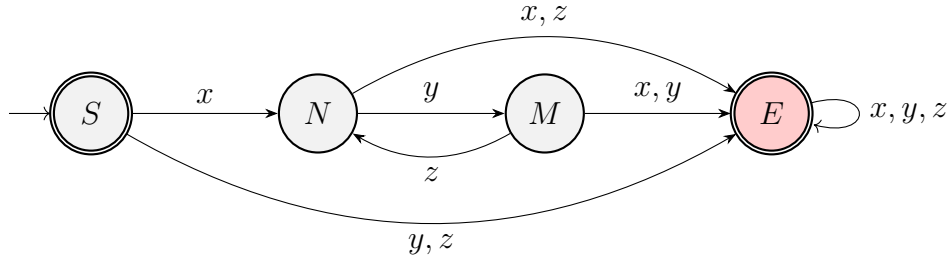
Sea $G' = (\{S, N, M, F\}, \{x, y, z\}, P, F)$ con P :

$$\begin{aligned} F &\rightarrow N \mid M \\ N &\rightarrow Sx \mid Mz \\ M &\rightarrow Ny \\ S &\rightarrow \varepsilon \end{aligned}$$

Tenemos que G' es una gramática regular por la izquierda, y $\mathcal{L}(G') = \mathcal{L}(G)$.

3. Encuentra el AFD que acepte el complementario del lenguaje $\mathcal{L}(G)$.

Intercambiando los estados finales por no finales y viceversa en el AFD de la Figura 1.17, obtenemos el AFD asociado al complementario del lenguaje $\mathcal{L}(G)$, mostrado en la Figura 1.18.

Figura 1.18: AFD asociado al lenguaje $\overline{\mathcal{L}(G)}$ del ejercicio 1.1.12.

q_1	×		
q_2	×	×	
q_3	×	(q_0, q_3)	×
	q_0	q_1	q_2

Tabla 1.3: Tabla de minimización de M_1 .

Ejercicio 1.1.13. Determinar autómatas minimales para los lenguajes $L(M_1) \cup L(M_2)$ y $L(M_1) \cap \overline{L(M_2)}$ donde,

- $M_1 = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \delta_1, q_0, \{q_2\})$ donde

δ_1	q_0	q_1	q_2	q_3
a	q_1	q_1	q_3	q_3
b	q_2	q_1	q_1	q_3
c	q_3	q_3	q_0	q_3

- $M_2 = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \delta_2, q_0, \{q_2\})$ donde

δ_2	q_0	q_1	q_2	q_3
a	q_1	q_1	q_3	q_3
b	q_1	q_2	q_2	q_3
c	q_3	q_3	q_0	q_3

En primer lugar, minimizamos M_1 y M_2 . La minimización de M_1 se muestra en la Tabla 1.3.

Por tanto, $\{q_1, q_3\}$ son indistinguibles, por lo que el AFD minimal asociado a M_1 es $M_1^m = (\{q_0, q_1, q_2\}, \{a, b, c\}, \delta_1^m, q_0, \{q_2\})$ donde:

δ_1^m	q_0	q_1	q_2
a	q_1	q_1	q_1
b	q_2	q_1	q_1
c	q_1	q_1	q_0

La minimización de M_2 se muestra en la Tabla 1.4.

Por tanto, el AFD minimal asociado a M_2 es $M_2^m = M_2$ minimal. A continuación, construimos los autómatas finitos deterministas asociados a $L(M_1) \cup L(M_2)$ y $L(M_1) \cap \overline{L(M_2)}$.

q_1	×		
q_2	×	×	
q_3	×	×	×
	q_0	q_1	q_2

Tabla 1.4: Tabla de minimización de M_2 .

δ	(q_0, q'_0)	(q_0, q'_1)	(q_0, q'_2)	(q_0, q'_3)	(q_1, q'_0)	(q_1, q'_1)	(q_1, q'_2)	(q_1, q'_3)	(q_2, q'_0)	(q_2, q'_1)	(q_2, q'_2)	(q_2, q'_3)
a	$q_1 q'_1$	$q_1 q'_1$	$q_1 q'_3$	$q_1 q'_3$	$q_1 q'_1$	$q_1 q'_1$	$q_1 q'_3$	$q_1 q'_3$	$q_1 q'_1$	$q_1 q'_1$	$q_1 q'_3$	$q_1 q'_3$
b	$q_2 q'_1$	$q_2 q'_2$	$q_2 q'_2$	$q_2 q'_3$	$q_1 q'_1$	$q_1 q'_2$	$q_1 q'_2$	$q_1 q'_3$	$q_1 q'_1$	$q_1 q'_2$	$q_1 q'_2$	$q_1 q'_3$
c	$q_1 q'_3$	$q_1 q'_3$	$q_1 q'_0$	$q_1 q'_3$	$q_1 q'_3$	$q_1 q'_3$	$q_1 q'_0$	$q_1 q'_3$	$q_0 q'_3$	$q_0 q'_3$	$q_0 q'_0$	$q_0 q'_3$

Tabla 1.5: Transiciones del autómata producto para $L(M_1)$, $L(M_2)$.1. $L(M_1) \cup L(M_2)$

En primer lugar, construimos el AFD asociado a $L(M_1) \cup L(M_2)$. Sea este $M = (Q, \{a, b, c\}, \delta, (q_0, q'_0), F)$, donde:

- $Q = \{q_i q'_j \mid i = 0, 1, 2 \text{ y } j = 0, 1, 2, 3\}$.
- $F = \{(q_2, q'_j) \mid j = 0, 1, 2, 3\} \cup \{(q_i, q'_2) \mid i = 0, 1, 2\}$.
- δ viene dada por la Tabla 1.5.

Los estados accesibles son:

$$\{q_0 q'_0, q_0 q'_3, q_1 q'_0, q_1 q'_1, q_1 q'_2, q_1 q'_3, q_2 q'_1, q_2 q'_3\}$$

La minimización de M se muestra en la Tabla 1.6.

Por tanto, el AFD minimal asociado a $L(M_1) \cup L(M_2)$ es $M^m = M$, el cual se puede ver en la Figura 1.19.

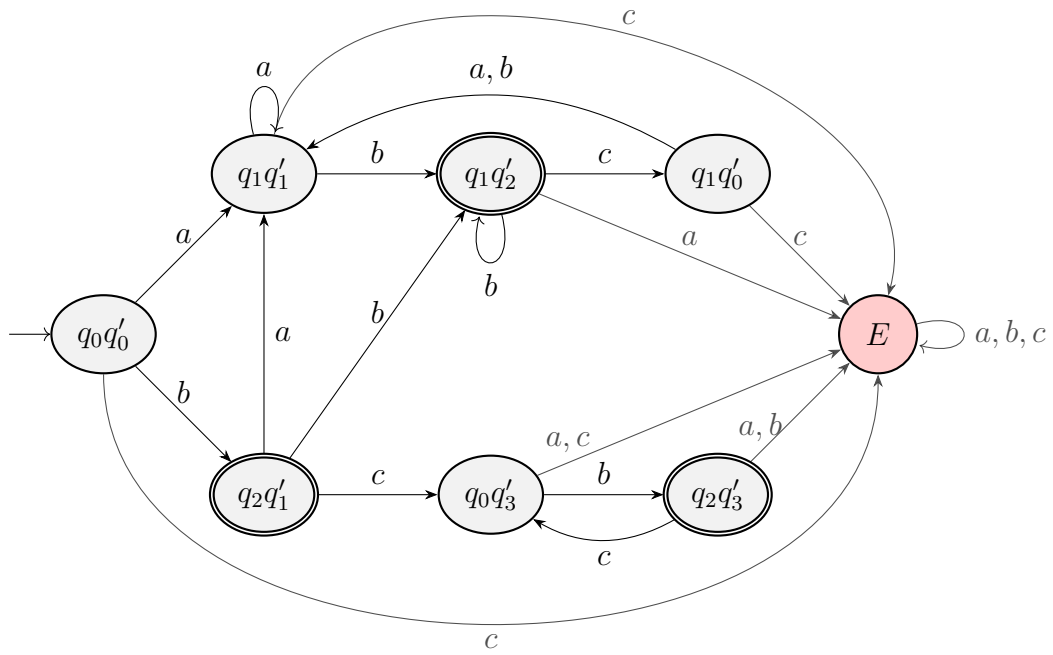
2. $L(M_1) \cap \overline{L(M_2)}$

Ya tenemos del apartado anterior el AFD minimal asociado a $L(M_1)$. La minimización de $\overline{L(M_2)}$ se muestra en la Tabla 1.7.

Por tanto, el AFD minimal asociado a $\overline{L(M_2)}$ es ya minimal. A continuación, construimos el AFD asociado a $L(M_1) \cap \overline{L(M_2)}$. Sea este $M = (Q, \{a, b, c\}, \delta, (q_0, q'_0), F)$, donde:

$q_0 q'_3$	×						
$q_1 q'_0$	×	×					
$q_1 q'_1$	×	×	×				
$q_1 q'_2$	×	×	×	×			
$q_1 q'_3$	×	×	×	×	×		
$q_2 q'_1$	×	×	×	×	×	×	
$q_2 q'_3$	×	×	×	×	×	×	×
	$q_0 q'_0$	$q_0 q'_3$	$q_1 q'_0$	$q_1 q'_1$	$q_1 q'_2$	$q_1 q'_3$	$q_2 q'_1$

Tabla 1.6: Tabla de minimización de M para $L(M_1) \cup L(M_2)$.

Figura 1.19: AFD minimal asociado a $L(M_1) \cup L(M_2)$.

q_1	×		
q_2	×	×	
q_3	×	×	×
	q_0	q_1	q_2

Tabla 1.7: Tabla de minimización de $\overline{M_2}$.

$q_0q'_3$							
$q_1q'_0$	×	×					
$q_1q'_1$	×	×					
$q_1q'_2$	×	×		$(q_1q'_2, q_1q'_0)$ $(q_1q'_1, q_1q'_0)$			
$q_1q'_3$	×	×	$(q_1q'_2, q_1q'_3)$ $(q_1q'_2, q_1q'_0)$ $(q_1q'_1, q_1q'_2)$ $(q_0q'_3, q_1q'_0)$	$(q_2q'_3, q_2q'_1)$ $(q_1q'_3, q_1q'_0)$ $(q_1q'_1, q_1q'_2)$	$(q_2q'_3, q_2q'_1)$ $(q_1q'_1, q_1q'_3)$		
$q_2q'_1$	×	×	×	×	×	×	
$q_2q'_3$	×	×	×	×	×	×	$(q_0q'_3, q_0q'_0)$
	$q_0q'_0$	$q_0q'_3$	$q_1q'_0$	$q_1q'_1$	$q_1q'_2$	$q_1q'_3$	$q_2q'_1$

Tabla 1.8: Tabla de minimización de M para $L(M_1) \cap \overline{L(M_2)}$.

δ^m	q_0	q_1	q_2
a	q_1	q_1	q_1
b	q_2	q_1	q_1
c	q_1	q_1	q_0

Tabla 1.9: Transiciones del autómata minimal para $L(M_1), \overline{L(M_2)}$.

- $Q = \{q_iq'_j \mid i = 0, 1, 2 \text{ y } j = 0, 1, 2, 3\}$.
- $F = \{(q_2, q'_j) \mid j = 0, 1, 3\}$.
- δ viene dada por la Tabla 1.5.

Los estados accesibles son los mismos que antes:

$$\{q_0q'_0, q_0q'_3, q_1q'_0, q_1q'_1, q_1q'_2, q_1q'_3, q_2q'_1, q_2q'_3\}$$

La minimización de M se muestra en la Tabla 1.8.

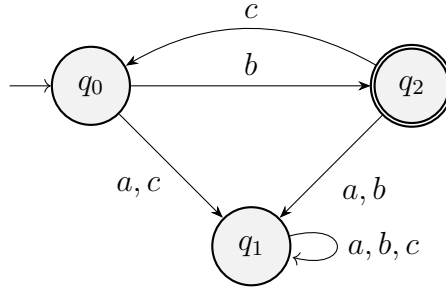
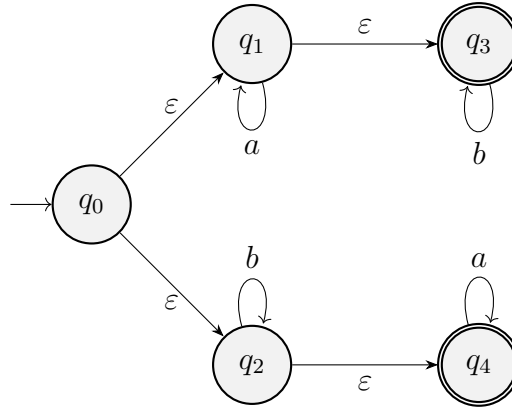
Por tanto, notando por \equiv a la relación de indistinguibilidad, tenemos que:

$$q_0q'_3 \equiv q_0q'_0 := q_0 \quad q_1q'_1 \equiv q_1q'_0 \equiv q_1q'_2 \equiv q_1q'_3 := q_1 \quad q_2q'_1 \equiv q_2q'_3 := q_2$$

Por tanto, el AFD minimal asociado a $L(M_1) \cap \overline{L(M_2)}$ es $M^m = (Q, \{a, b, c\}, \delta^m, q_0, F)$ donde:

- $Q = \{q_0, q_0q'_1, q_0q'_2, q_1, q_2, q_2q'_0, q_2q'_2\}$.
- $F = \{q_2, q_2q'_0\}$.
- Los estados accesibles son $\{q_0, q_1, q_2\}$.
- δ^m viene dada por la Tabla 1.9, donde solo la definimos para los estados accesibles.

El AFD minimal asociado a $L(M_1) \cap \overline{L(M_2)}$ es el de la Figura 1.20.

Figura 1.20: AFD minimal asociado a $L(M_1) \cap \overline{L(M_2)}$.Figura 1.21: AFND con transiciones nulas asociado a la expresión regular $a^*b^* + b^*a^*$.

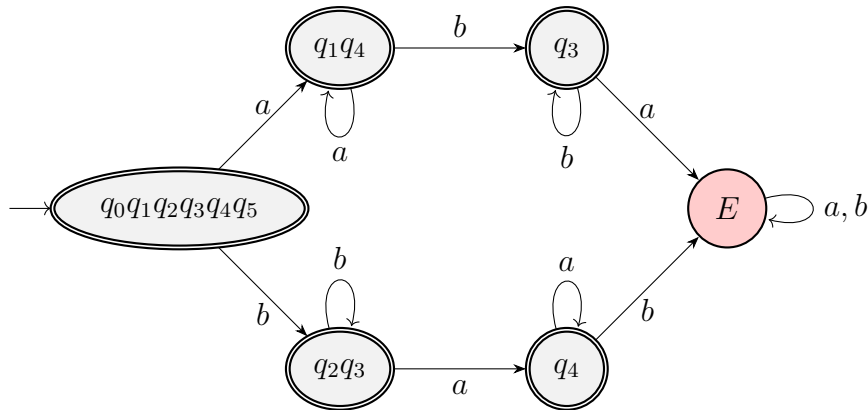
Ejercicio 1.1.14. Dado el conjunto regular representado por la expresión regular $a^*b^* + b^*a^*$, construir un autómata finito determinístico minimal que lo acepte.

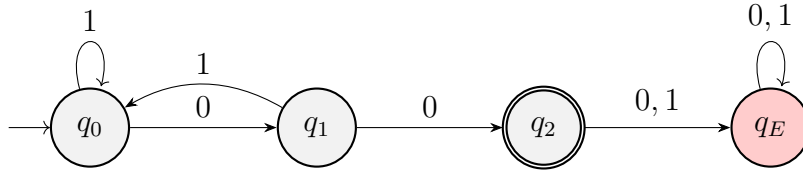
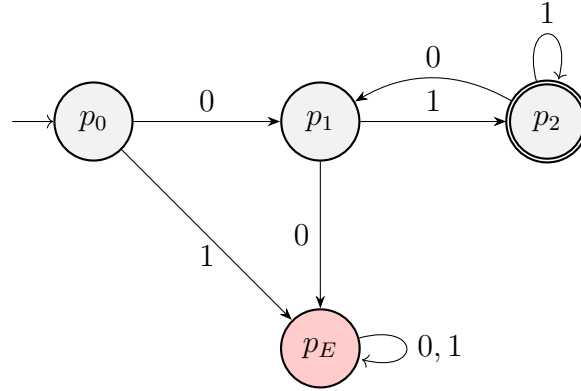
El AFND con transiciones nulas asociado a la expresión regular dada es el de la Figura 1.21.

El AFD asociado a la expresión regular dada es el de la Figura 1.22.

Veamos ahora que el AFD de la Figura 1.22 es minimal.

- El estado E es distinguible de cualquier otro estado pues ser el único estado que no es final.

Figura 1.22: AFD asociado a la expresión regular $a^*b^* + b^*a^*$.

Figura 1.23: AFD minimal asociado a $L_1 = (01 + 1)^*00$.Figura 1.24: AFD minimal asociado a $L_2 = 01(01 + 1)^*$.

- Veamos que q_3 es distinguible del resto de estados finales. Leyendo a , tenemos que:

$$\delta(q_3, a) = E \notin F, \quad \begin{cases} \delta(q_1q_4, a) = q_1q_4 \in F, \\ \delta(q_2q_3, a) = q_4 \in F, \\ \delta(q_4, a) = q_4 \in F, \\ \delta(q_0q_1q_2q_3q_4q_5, a) = q_1q_4 \in F. \end{cases}$$

- De forma análoga leyendo b , tenemos que q_4 es distinguible del resto de estados finales.
- El estado q_1q_4 es distinguible de q_2q_3 y $q_0q_1q_2q_3q_4q_5$ leyendo ba .
- Finalmente, $q_0q_1q_2q_3q_4q_5$ es distinguible de q_2q_3 leyendo ab .

Ejercicio 1.1.15. Sean los lenguajes:

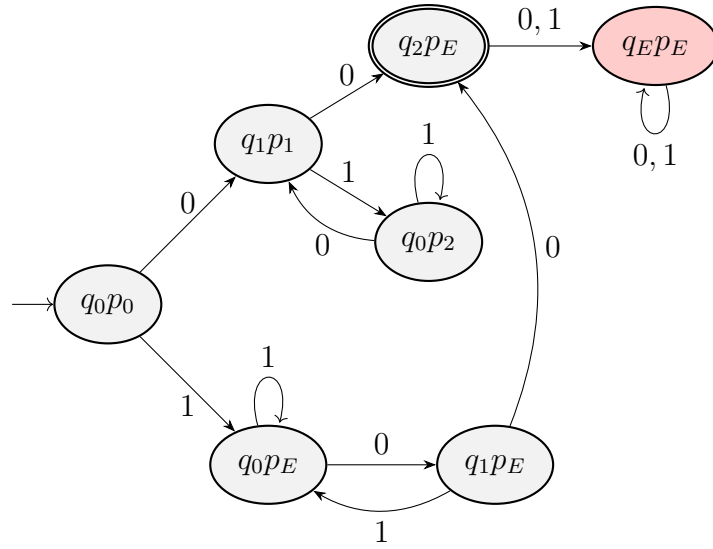
1. $L_1 = (01 + 1)^*00$
2. $L_2 = 01(01 + 1)^*$

construir un autómata finito determinístico minimal que acepte el lenguaje $L_1 \setminus L_2$, a partir de autómatas que acepten L_1 y L_2 .

Veamos que $L_1 \cap L_2 = \emptyset$. Sea $z \in L_1$. Entonces, z es admitida por la expresión regular $(01 + 1)^*00$, por lo que termina en 0. Por tanto, $z \notin L_2$, ya que todas las palabras de L_2 terminan por 1. Por tanto, $L_1 \cap L_2 = \emptyset$ y $L_1 \setminus L_2 = L_1$. Aun así, haremos el proceso algorítmico para obtener el AFD minimal asociado a $L_1 \setminus L_2$.

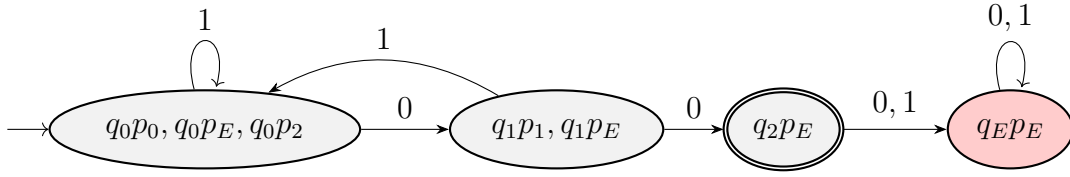
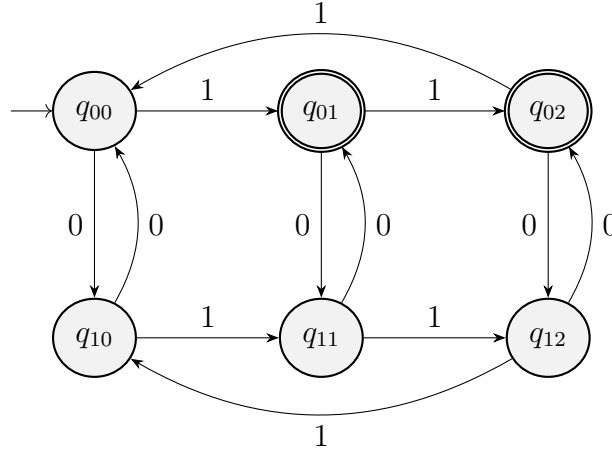
El AFD asociado a L_1 es el de la Figura 1.23.

El AFD asociado a L_2 es el de la Figura 1.24.

Figura 1.25: AFD asociado a $L_1 \setminus L_2$.

q_1p_1	×					
q_2p_E	×	×				
q_0p_2		×	×			
q_0p_E		×	×	(q_1p_E, q_1p_1) (q_0p_0, q_0p_2)		
q_1p_E	×	(q_0p_E, q_0p_0) (q_0p_E, q_0p_2)	×	×	×	
q_Ep_E	×	×	×	×	×	×
	q_0p_0	q_1p_1	q_2p_E	q_0p_2	q_0p_E	q_1p_E

Tabla 1.10: Tabla de minimización de $L_1 \setminus L_2$.

Figura 1.26: AFD minimal asociado a $L_1 \setminus L_2$.Figura 1.27: AFD que acepta el lenguaje L del Ejercicio 1.1.16.

El AFD asociado a $L_1 \setminus L_2$ es el de la Figura 1.25.

La minimización del AFD de la Figura 1.25 se muestra en la Tabla 1.10.

Por tanto, notando por \equiv a la relación de indistinguibilidad, tenemos que:

$$q_0p_0 \equiv q_0p_E \equiv q_0p_2 \quad q_1p_1 \equiv q_1p_E$$

El AFD minimal asociado a $L_1 \setminus L_2$ es el de la Figura 1.26. Como vemos, este autómata es isomorfo al autómata de la Figura 1.23 que aceptaba L_1 , como ya habíamos predicho.

Ejercicio 1.1.16. Dados los alfabetos $A = \{0, 1, 2, 3\}$ y $B = \{0, 1\}$ y el homomorfismo f de A^* en B^* dado por:

$$f(0) = 00, \quad f(1) = 01, \quad f(2) = 10, \quad f(3) = 11$$

Sea L el conjunto de las palabras de B^* en las que el número de símbolos 0 es par y el de símbolos 1 no es múltiplo de 3. Construir un autómata finito determinista que acepte el lenguaje $f^{-1}(L)$.

El AFD que acepta a L se desarrolló en el Ejercicio ???. Se muestra no obstante de nuevo en la Figura 1.27.

El AFD que acepta a $f^{-1}(L)$ es el de la Figura 1.28.

Ejercicio 1.1.17. Determinar un autómata finito determinístico minimal para el lenguaje sobre el alfabeto $A = \{a, b, c\}$ dado por la expresión regular $b(a + b)^* + cb^*$.

El AFD asociado a la expresión regular dada es el de la Figura 1.29.

Veamos que es minimal:

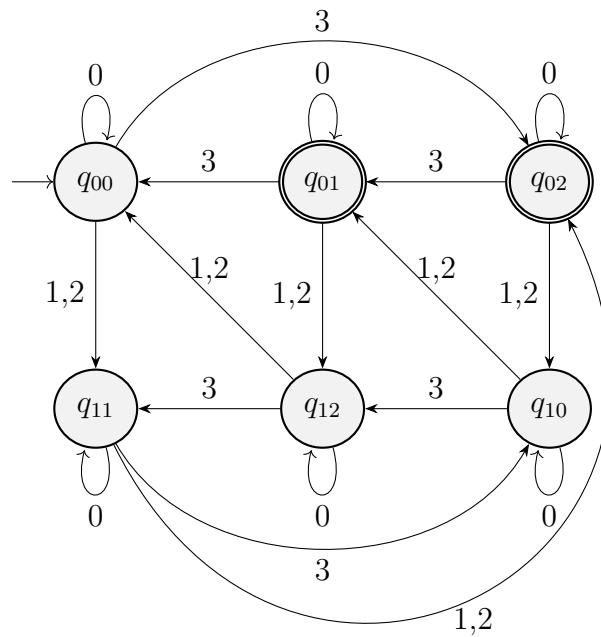


Figura 1.28: AFD que acepta el lenguaje L del Ejercicio 1.1.16.

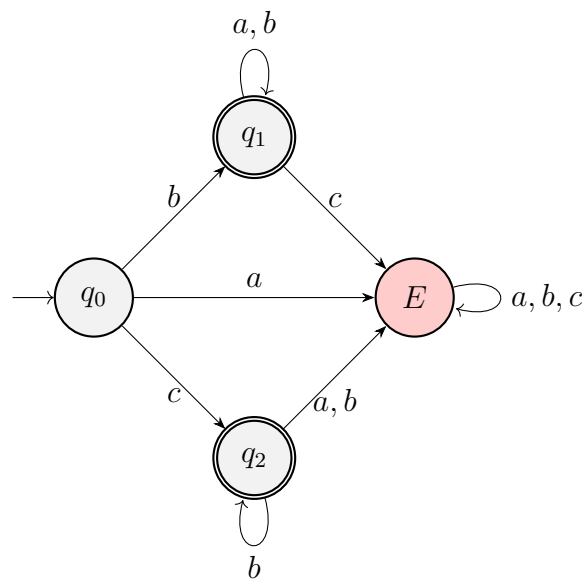


Figura 1.29: AFD asociado a la expresión regular $b(a + b)^* + cb^*$.

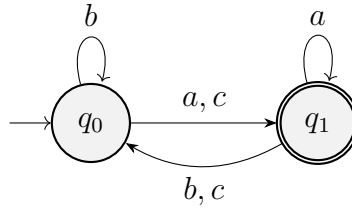
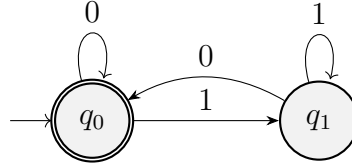
Figura 1.30: AFD minimal asociado a $L_1 = L_3$ del Ejercicio 1.1.18.

Figura 1.31: AFD minimal asociado a los múltiplos de 2 del Ejercicio 1.1.19.

- E es distinguible de cualquier otro estado puesto que desde él no se puede llegar a un estado final.
- q_0 es distinguible de q_1 y q_2 puesto que no es final.
- q_1 es distinguible de q_2 leyendo a .

Ejercicio 1.1.18. Determinar si las expresiones regulares siguientes representan el mismo lenguaje:

1. $L_1 = (b + (c + a)a^*(b + c))^*(c + a)a^*$
2. $L_2 = b^*(c + a)((b + c)b^*(c + a))^*a^*$
3. $L_3 = b^*(c + a)(a^*(b + c)b^*(c + a))^*a^*$

Justificar la respuesta.

Veamos en primer lugar que $L_2 \neq L_1, L_3$. Sea la palabra $u = caaccaaa$. Tenemos que $u \in L_1, L_3$ pero $u \notin L_2$. Por tanto, $L_2 \neq L_1, L_3$.

Veamos ahora que $L_1 = L_3$ obteniendo el autómata finito minimal asociado a cada uno de ellos y viendo que es igual. Este se encuentra en la Figura 1.30.

Ejercicio 1.1.19. Construir un autómata finito determinista minimal que acepte el conjunto de palabras sobre el alfabeto $A = \{0, 1\}$ que representen números no divisibles por dos ni por tres (en binario).

El AFD asociado a los múltiplos de 2 se muestra en la Figura 1.31.

El AFD asociado a los múltiplos de 3 se muestra en la Figura 1.32.

El autómata descrito en el enunciado es el autómata producto de los complementarios a los dos anteriores, mostrado en la Figura 1.33.

La minimización del autómata de la Figura 1.33 se muestra en la Tabla 1.11.

Por tanto, notando por \equiv a la relación de indistinguibilidad, tenemos que:

$$q_0 p_0 \equiv q_1 p_0 \equiv p_0$$

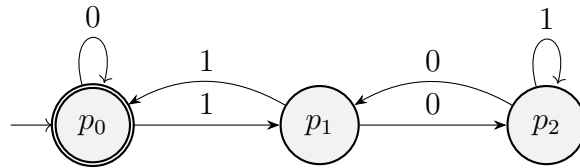


Figura 1.32: AFD minimal asociado a los múltiplos de 3 del Ejercicio 1.1.19.

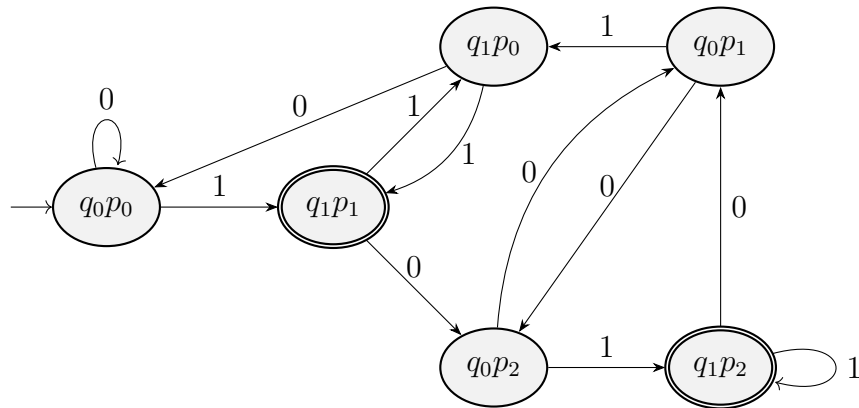


Figura 1.33: AFD asociado al lenguaje del Ejercicio 1.1.19.

q_1p_1	×				
q_1p_0		×			
q_0p_2	×	×	×		
q_0p_1	×	×	×	×	
q_1p_2	×	×	×	×	×
	q_0p_0	q_1p_1	q_1p_0	q_0p_2	q_1p_1

Tabla 1.11: Tabla de minimización del autómata de la Figura 1.33.

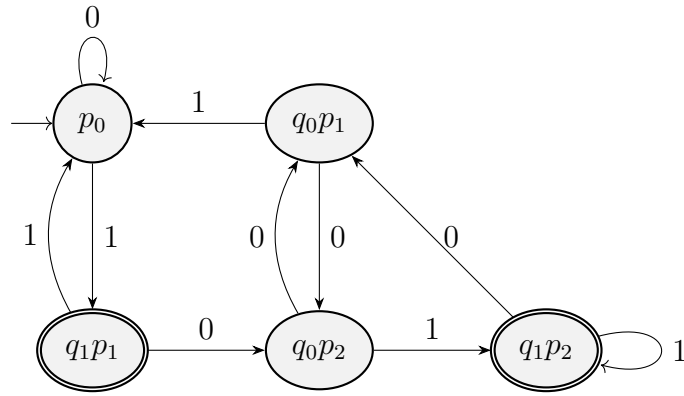


Figura 1.34: AFD Minimal asociado al lenguaje del Ejercicio 1.1.19.

Por tanto, el autómata minimal asociado al lenguaje del enunciado es el de la Figura 1.34.

Ejercicio 1.1.20. Determinar una expresión regular para los siguientes lenguajes sobre el alfabeto $\{0, 1\}$:

- Palabras en las que el tercer símbolo es un 0.
- Palabras en las que el antepenúltimo símbolo es un 1.

Construir un autómata finito minimal que acepte la intersección de ambos lenguajes.

Respecto al lenguaje de las palabras en las que el tercer símbolo es un 0 (llamémoslo L_1), su expresión regular es:

$$(0 + 1)(0 + 1) \text{ 0 } (0 + 1)^*$$

Respecto al lenguaje de las palabras en las que el antepenúltimo símbolo es un 1 (llamémoslo L_2), su expresión regular es:

$$(0 + 1)^* \text{ 1 } (0 + 1)(0 + 1)$$

Opción 1: Autómata Producto.

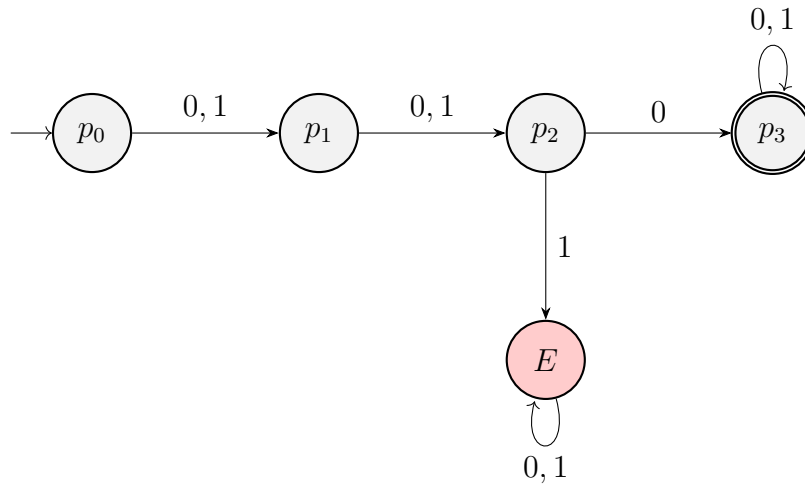
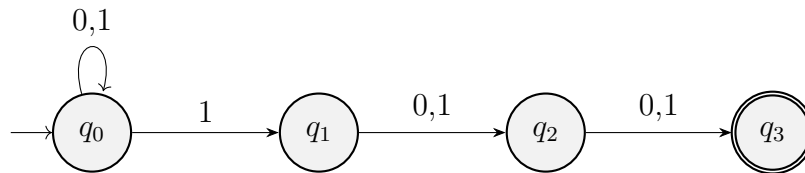
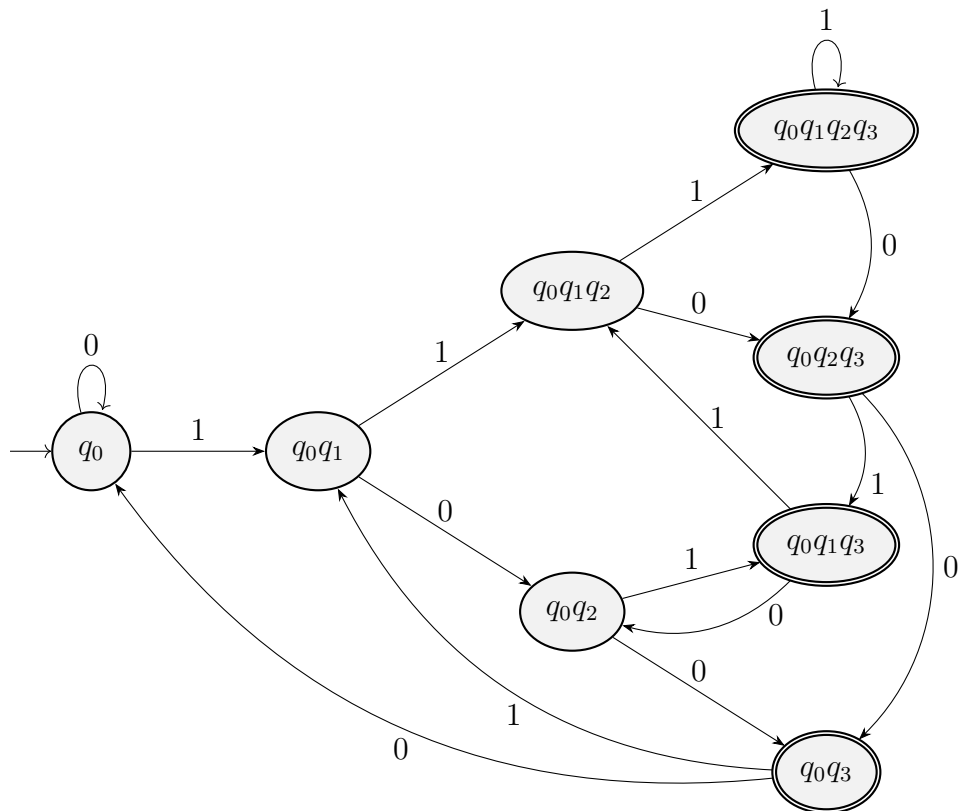
Obtendremos el AFD de L_1 de forma directa, mostrado en la Figura 1.35.

Podríamos optar con construir el AFD de L_2 de forma directa, pero construiremos un AFND que acepte el lenguaje y luego lo convertiremos a AFD. El AFND se muestra en la Figura 1.36, cuyos estados son:

- q_0 : No estamos en la cadena final, por lo que podemos leer 0's y 1's.
- q_1 : Acabo de empezar la cadena final. He leído un 1.
- q_2 : Estoy en la cadena final. El leído el 1 y el segundo símbolo.
- q_3 : Hemos terminado la cadena final.

Convertimos ahora el AFND de la Figura 1.36 en un AFD, representado en la Figura 1.37.

El AFD de la Figura 1.37 acepta el lenguaje L_2 , y es idéntico al que podríamos haber razonado de forma directa. Veamos qué representa cada estado:

Figura 1.35: AFD minimal que acepta el lenguaje L_1 del ejercicio 1.1.20.Figura 1.36: AFND que acepta el lenguaje L_2 del ejercicio 1.1.20.Figura 1.37: AFD que acepta el lenguaje L_2 del ejercicio 1.1.20.

q_0q_1	×						
$q_0q_1q_2$	×	$(q_0q_3, q_0q_1q_3)$					
q_0q_2	$(q_0q_3, q_0q_1q_3)$	×	×				
$q_0q_1q_2q_3$	×	×	×	×			
$q_0q_2q_3$	×	×	×	×	×		
$q_0q_1q_3$	×	×	×	×	×	×	
q_0q_3	×	×	×	×	×	×	×
	q_0	q_0q_1	$q_0q_1q_2$	q_0q_2	$q_0q_1q_2q_3$	$q_0q_2q_3$	$q_0q_1q_3$

Tabla 1.12: Tabla de minimización del autómata de la Figura 1.37.

- q_0 : No estamos en un candidato a ser cadena final. Si leemos un 1, empezaremos la que puede ser la cadena final.
- q_0q_1 : Hemos leído un 1, por lo que hemos empezado la posible cadena final. Llevamos 1.
- $q_0q_1q_2$: Hemos leído un 1 y un 1. Llevamos 11 de cadena final.
- q_0q_2 : Hemos leído un 1 y un 0. Llevamos 10 de cadena final.
- $q_0q_1q_2q_3$: Hemos leído un 1, un 1 y un 1. Llevamos 111 de cadena final.
- $q_0q_2q_3$: Hemos leído un 1, un 1 y un 0. Llevamos 110 de cadena final.
- $q_0q_1q_3$: Hemos leído un 1, un 0 y un 1. Llevamos 101 de cadena final.
- q_0q_3 : Hemos leído un 1, un 0 y un 0. Llevamos 100 de cadena final.

Lo complejo de hacerlo de forma directa sería ver las transiciones desde los estados finales. Razonando cuál es la cadena final leída, podríamos haberlo hecho de forma directa, pero el AFND nos ha ayudado a hacerlo de forma algorítmica. La minimización del autómata de la Figura 1.37 se muestra en la Tabla 1.12, donde vemos que este era minimal.

Como vemos, obtener el autómata producto será complejo, puesto que tendrá $8 \cdot 5 = 40$ estados. Por tanto, optamos por la segunda opción.

Opción 2: Expresión Regular.

La expresión regular del lenguaje intersección ha de ser la siguiente:

$$(0 + 1)(0 + 1) \text{ 0 } (0 + 1)^* \text{ 1 } (0 + 1)(0 + 1)$$

Observación. Notemos que esta expresión regular no contempla las palabras de longitud menor o igual a 5. Estudiemos estas:

- Si $|z| = 5$, es necesario que el tercer símbolo sea un 0 y el antepenúltimo ($5 - 2 = 3$) un 1. Por tanto, como el tercer símbolo no puede tomar ambos valores a la vez, no hay palabras de longitud 5.
- Si $|z| = 4$, es necesario que el tercer símbolo sea un 0 y el antepenúltimo ($4 - 2 = 2$) un 1. Por tanto, estas palabras son de la forma $(0+1) \text{ 10 } (0+1)$.
- Si $|z| = 3$, es necesario que el tercer símbolo sea un 0 y el antepenúltimo ($3 - 2 = 1$) un 1. Por tanto, estas palabras son de la forma $\text{1 } (0 + 1) \text{ 0}$.

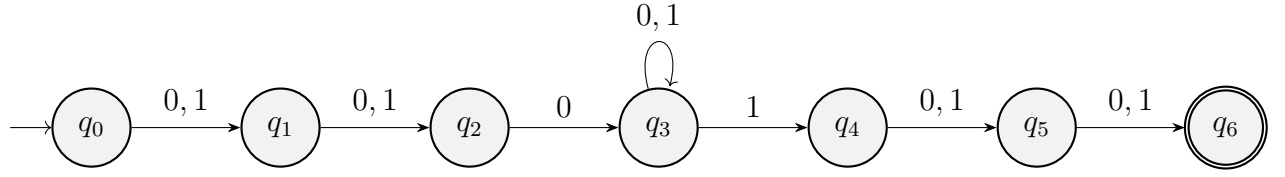


Figura 1.38: AFND que acepta la intersección de los lenguajes del ejercicio 1.1.20.

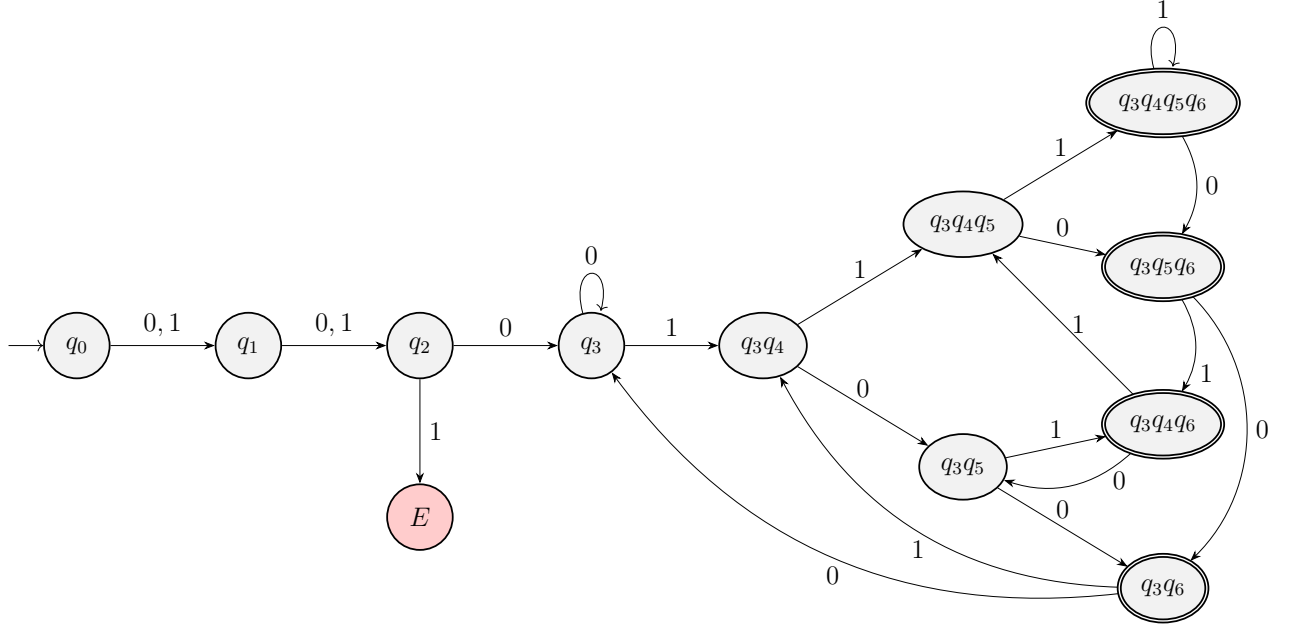


Figura 1.39: AFD que acepta la intersección de los lenguajes del ejercicio 1.1.20.

- Si $|z| < 3$, no podemos considerar el tercer símbolo, por lo que no hay palabras de longitud menor que 3.

Por tanto, la expresión regular correcta sería:

$$(0 + 1)(0 + 1) \text{ 0 } (0 + 1)^* \text{ 1 } (0 + 1)(0 + 1) + (0 + 1) \text{ 10 } (0 + 1) + \text{ 1 } (0 + 1) \text{ 0 }$$

No obstante, obtener el autómata finito minimal asociado a esta expresión regular sería muy complejo, por lo que no consideramos dichas palabras.

El AFND que acepta el lenguaje intersección se muestra en la Figura 1.38.

Convertimos ahora el AFND de la Figura 1.38 en un AFD, representado en la Figura 1.39.

Ejercicio 1.1.21. Construir un autómata finito minimal para los siguientes lenguajes sobre el alfabeto $\{0, 1\}$:

1. Palabras que contienen como subcadena una palabra del conjunto $\{00, 11\}^2$.
Han de contener una subcadena del conjunto $\{0000, 0011, 1100, 1111\}$. Para ello, construimos el autómata de la Figura 1.40.
La minimización del autómata de la Figura 1.40 se muestra en la Tabla 1.13.
Por tanto, tenemos que todos los estados finales eran indistinguibles. Notándolos por q_F , el autómata minimal es el de la Figura 1.41.

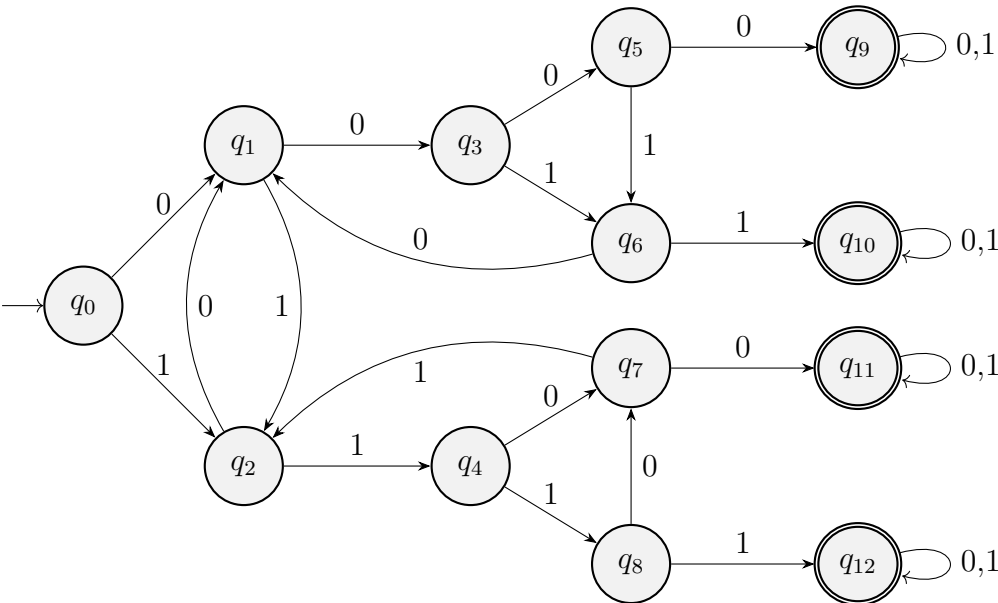


Figura 1.40: AFD que acepta el lenguaje del ejercicio 1.1.21.1.

q_1	×											
q_2	×	×										
q_3	×	×	×									
q_4	×	×	×	×								
q_5	×	×	×	×	×							
q_6	×	×	×	×	×	×						
q_7	×	×	×	×	×	×	×					
q_8	×	×	×	×	×	×	×	×				
q_9	×	×	×	×	×	×	×	×	×			
q_{10}	×	×	×	×	×	×	×	×	×	×		
q_{11}	×	×	×	×	×	×	×	×	×	×	×	
q_{12}	×	×	×	×	×	×	×	×	×	×	×	×
	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}

Tabla 1.13: Tabla de minimización del autómata de la Figura 1.40.

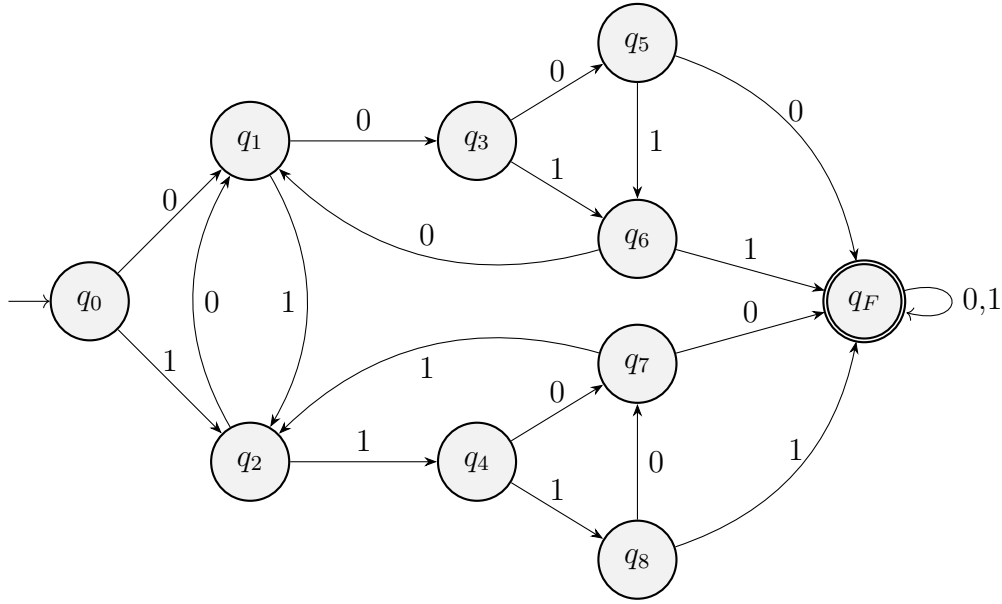


Figura 1.41: AFD minimal que acepta el lenguaje del ejercicio 1.1.21.1.

2. Palabras que contienen como subcadena una palabra del conjunto $\{0011, 1100\}$.

El AFD minimal que acepta este lenguaje es el de la Figura 1.42.

Ejercicio 1.1.22. Responda a los siguientes apartados:

1. Construye una gramática regular que genere el siguiente lenguaje:

$$L_1 = \{u \in \{0, 1\}^* \mid \text{el número de 1's y el número de 0's en } u \text{ es par} \}$$

Este es muy similar al Ejercicio ??. Tenemos los siguientes estados:

- \underline{S} : La cadena leída es correcta, ya que el número de 0's y de 1's es par.
- $\underline{E_0}$: Tenemos un error en 0, ya que el número de 0's es impar. El número de 1's es par.
- $\underline{E_1}$: Tenemos un error en 1, ya que el número de 1's es impar. El número de 0's es par.
- $\underline{E_{01}}$: Tenemos un error en 0 y en 1, ya que el número de 0's y de 1's es impar.

La gramática obtenida es $G = (\{E_{01}, E_0, E_1, S\}, \{0, 1\}, P, S)$, donde P es:

$$\begin{aligned} S &\rightarrow 0E_0 \mid 1E_1 \mid \varepsilon, \\ E_0 &\rightarrow 0S \mid 1E_{01}, \\ E_1 &\rightarrow 0E_{01} \mid 1S, \\ E_{01} &\rightarrow 0E_1 \mid 1E_0 \end{aligned}$$

El autómata finito determinista minimal asociado a la gramática obtenida es el de la Figura 1.43.

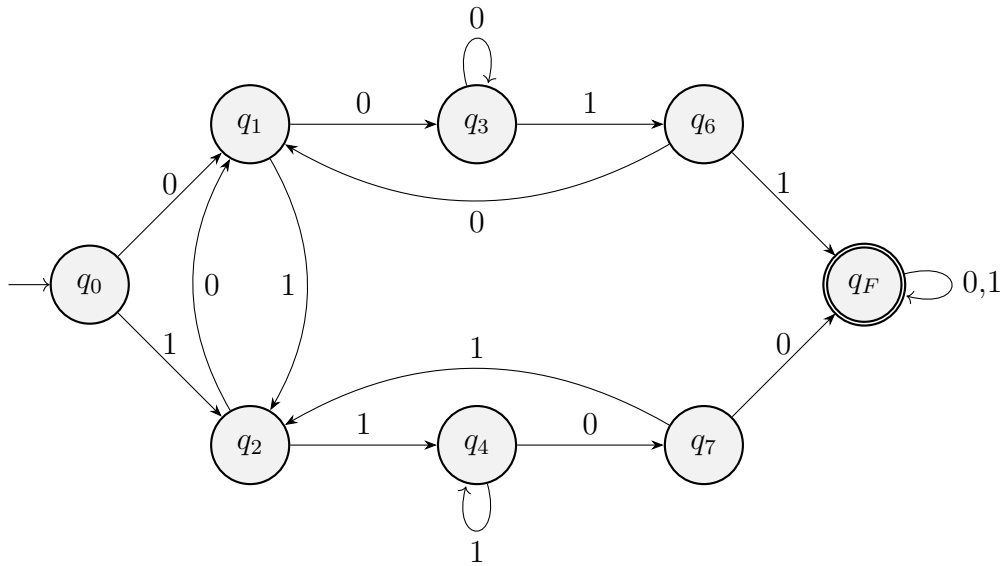


Figura 1.42: AFD minimal que acepta el lenguaje del ejercicio 1.1.21.2.

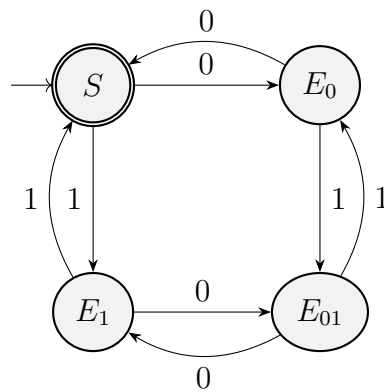


Figura 1.43: AFD minimal asociado a L_1 del Ejercicio 1.1.22.

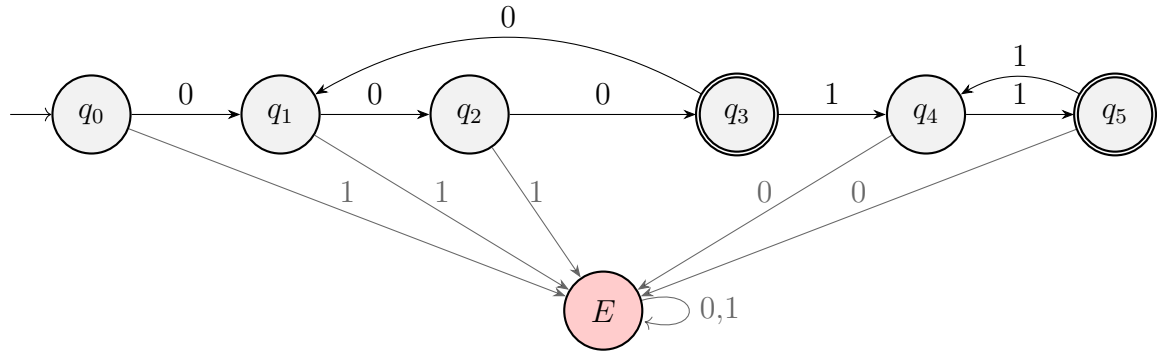


Figura 1.44: AFD minimal asociado a L_2 del Ejercicio 1.1.22.

2. Construye un autómata que reconozca el siguiente lenguaje:

$$L_2 = \{0^n 1^m \mid n \geq 1, m \geq 0, n \text{ múltiplo de } 3, m \text{ par} \}$$

Sean los siguientes estados:

- $\underline{q_0}$: $n, m = 0$.
- $\underline{q_1}$: $n \bmod 3 = 1, m = 0$.
- $\underline{q_2}$: $n \bmod 3 = 2, m = 0$.
- $\underline{q_3}$: $n \bmod 3 = 0, n > 1, m = 0$.
- $\underline{q_4}$: $n \bmod 3 = 0, m \bmod 2 = 1$.
- $\underline{q_5}$: $n \bmod 3 = 0, m \bmod 2 = 0$.

El autómata finito determinista asociado a L_2 es el de la Figura 1.44.

3. Diseña el AFD mínimo que reconoce el lenguaje $(L_1 \cup L_2)$.

Ejercicio 1.1.23. Sobre el alfabeto $\{0, 1\}$:

1. Construye una gramática regular que genere el lenguaje L_1 de las palabras u tales que:

- Si $|u| < 5$ entonces el número de 1's es impar.
- Si $|u| \geq 5$ entonces el número de 1's es par.
- u tiene al menos un símbolo 1.

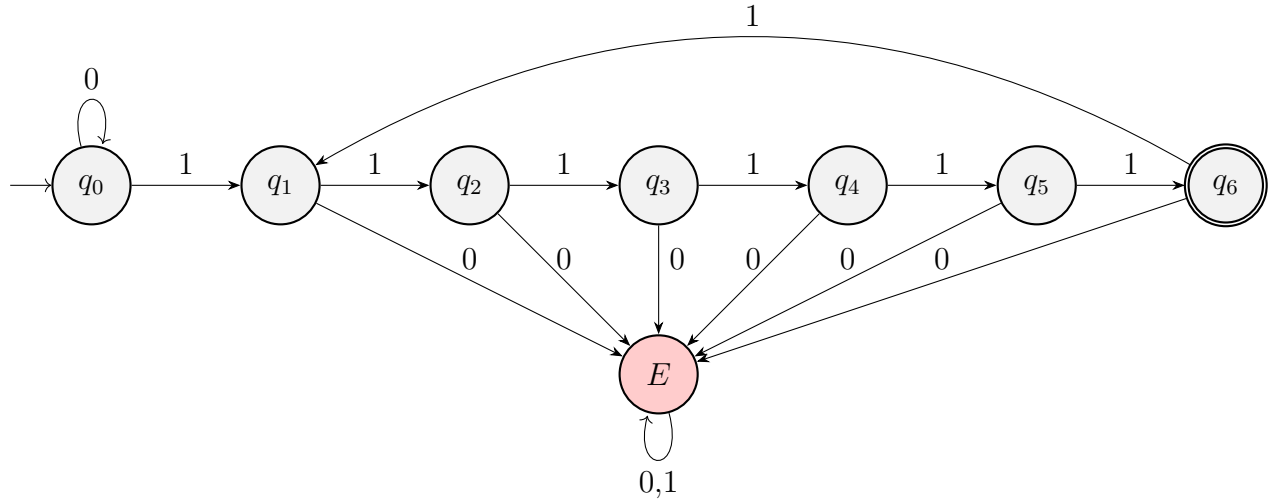
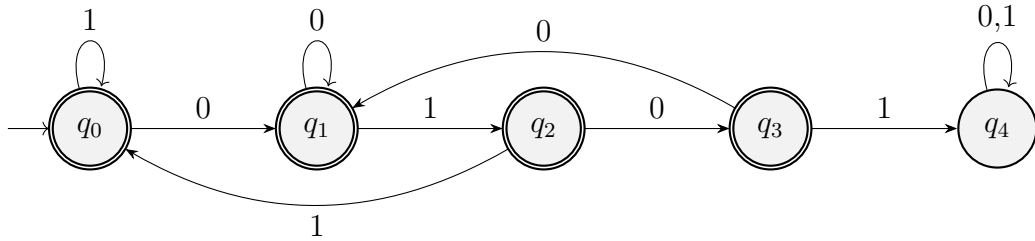
2. Construye un autómata que reconozca el lenguaje L_2 dado por:

$$L_2 = \{0^n 1^m \mid n \geq 0, m \geq 1, m \text{ es múltiplo de } 6\}$$

El autómata finito determinista asociado a L_2 es el de la Figura 1.45.

3. Diseña el AFD mínimo que reconozca el lenguaje $(L_1 \cup L_2)$.

Ejercicio 1.1.24. Encuentra para cada uno de los siguientes lenguajes una gramática de tipo 3 que lo genere o un autómata finito que lo reconozca:

Figura 1.45: AFD minimal asociado a L_2 del Ejercicio 1.1.23.2.Figura 1.46: AFD minimal asociado a L_1 del Ejercicio 1.1.24.

- $L_1 = \{u \in \{0, 1\}^* \mid u \text{ no contiene la subcadena "0101"}\}$.
- $L_2 = \{0^i 1^j 0^k \mid i \geq 1, k \geq 0, i \text{ impar}, k \text{ múltiplo de } 3 \text{ y } j \geq 2\}$.

Diseña el AFD mínimo que reconoce el lenguaje $(L_1 \cap L_2)$.

El AFD minimal asociado a L_1 es el de la Figura 1.46.

El AFD minimal asociado a L_2 es el de la Figura 1.47.

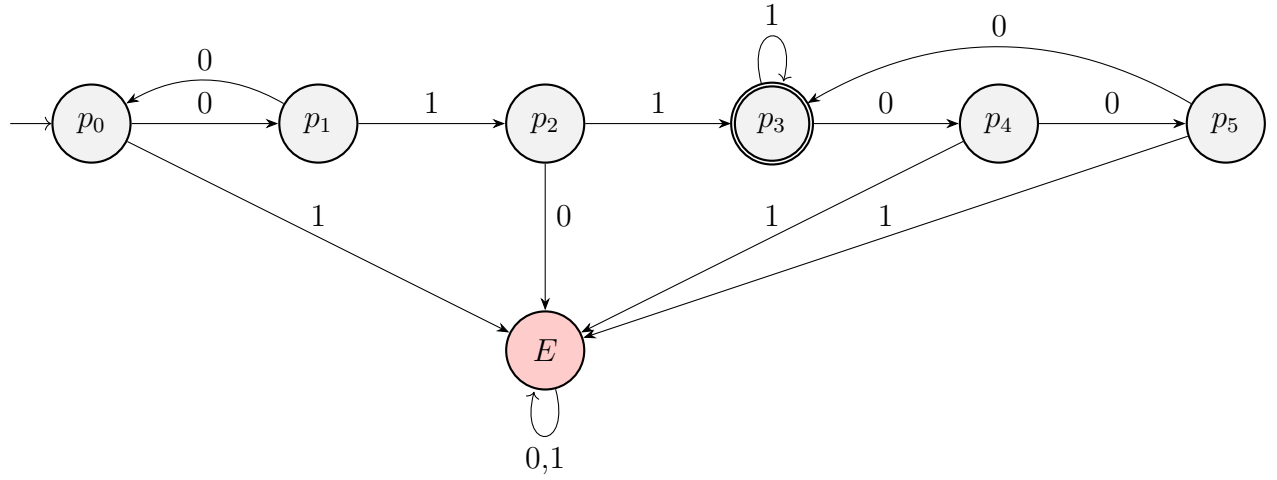
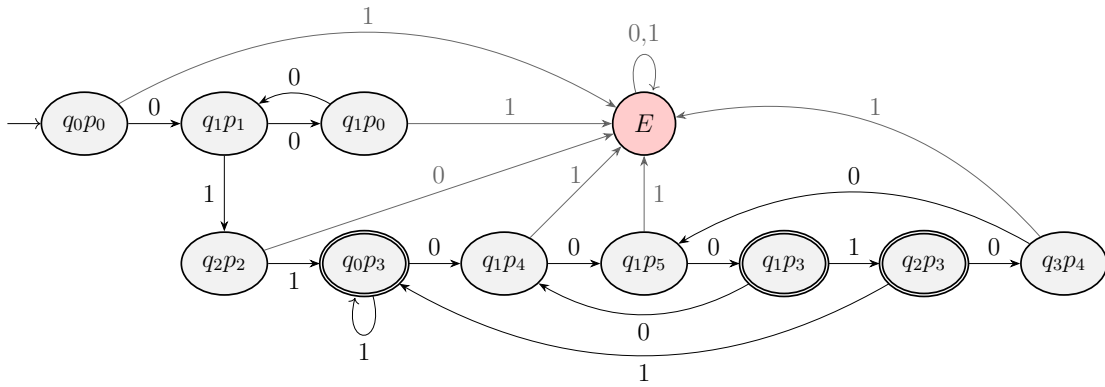
El AFD asociado a $(L_1 \cap L_2)$ es el de la Figura 1.48. Notemos que todos los estados de la forma q_i, E se han agrupado en un único estado E , ya que todos ellos son indistinguibles puesto que no se puede llegar desde ellos a ningún final.

La minimización de este autómata se muestra en la Tabla 1.14.

El AFD minimal asociado a $(L_1 \cap L_2)$ es el de la Figura 1.49. Como podemos ver en el AFD minimal de la Figura 1.49, este es isomorfo al de la Figura 1.47, por lo que $L_1 \cap L_2 = L_2$. Esto es algo que podríamos haber deducido al principio, ya que si $L_2 \subset L_1$.

Ejercicio 1.1.25. Dado el alfabeto $A = \{a, b, c\}$, encuentra:

1. Un AFD que reconozca las palabras en las que cada “c” va precedida de una “a” o una “b”.
2. Una expresión regular que represente el lenguaje compuesto por las palabras de longitud impar en las que el símbolo central es una “c”.

Figura 1.47: AFD minimal asociado a L_2 del Ejercicio 1.1.24.Figura 1.48: AFD asociado a $(L_1 \cap L_2)$ del Ejercicio 1.1.24.

q_1p_1	×									
q_1p_0		×								
q_2p_2	×	×	×							
q_0p_3	×	×	×	×						
q_1p_4	×	×	×	×	×					
q_1p_5	×	(q_0p_0, q_3p_4)	×	×	×	×				
q_1p_3	×	(q_1p_0, q_3p_4)	×	×		×	×			
q_2p_3	×	×	×	×	(q_1p_3, q_0p_3)	×	×	(q_1p_3, q_2p_3)		
q_3p_4	×	×	×	×	×	(q_0p_3, q_2p_3) (q_1p_3, q_2p_3)	×	×	×	
E	×	×	×	×	×	×	×	×	×	×
	q_0p_0	q_1p_1	q_1p_0	q_2p_2	q_0p_3	q_1p_4	q_1p_5	q_1p_3	q_2p_3	q_3p_4

Tabla 1.14: Tabla de minimización del autómata de la Figura 1.48.

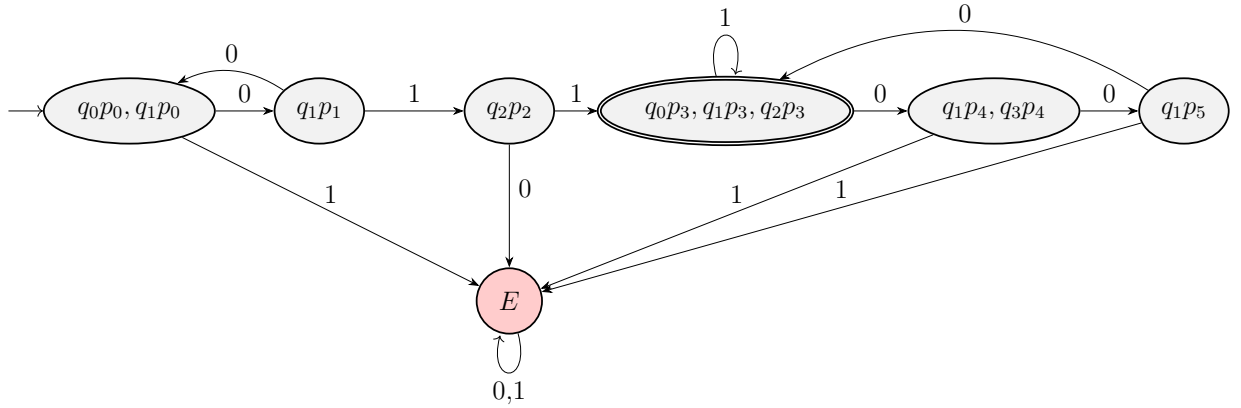
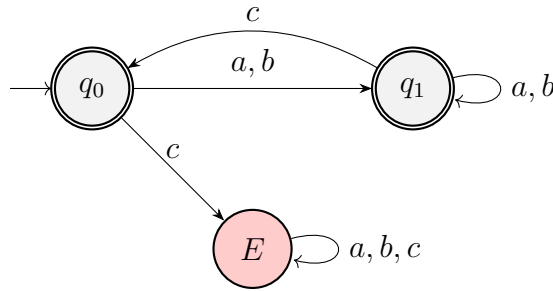
Figura 1.49: AFD minimal asociado a $(L_1 \cap L_2)$ del Ejercicio 1.1.24.

Figura 1.50: AFD minimal asociado al lenguaje del Ejercicio 1.1.25.1.

Veamos que este lenguaje no es regular usando el lema de bombeo. Para todo $n \in \mathbb{N}$, tomamos la palabra $z = a^n c a^n \in L$, con $|z| = 2n + 1 \geq n$. Toda descomposición de z en uvw , con $u, v, w \in \{0, 1\}^*$, $|uv| \leq n$ y $|v| \geq 1$ debe tener:

$$u = a^k, \quad v = a^l, \quad w = a^{n-k-l} c a^n \quad \text{con } l, k \in \mathbb{N} \cup \{0\}, l \geq 1, k + l \leq n$$

Para $i = 2$, tenemos que $uv^2w = a^{k+2l+n-k-l} c a^n = a^{n+l} c a^n \notin L$ ya que, como $l \geq 1$, $n + l \neq n$, por lo que c no es el símbolo central. Por lo tanto, por el contrarrecíproco del lema de bombeo, L no es regular; y por tanto no podemos encontrar una expresión regular que lo represente.

3. Una gramática regular que genere las palabras de longitud impar.

Sea $G = (\{S, X\}, A, P, S)$, donde P es:

$$\begin{aligned} S &\rightarrow aX \mid bX \mid cX, \\ X &\rightarrow aS \mid bS \mid cS \mid \varepsilon. \end{aligned}$$

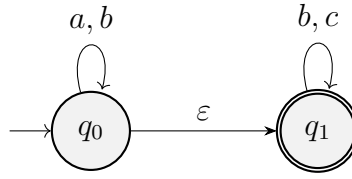
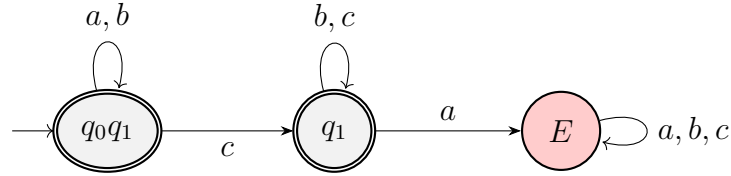
Tenemos que $\mathcal{L}(G) = \{w \in A^* \mid |w| \text{ es impar}\}$.

Ejercicio 1.1.26. Construir autómatas finitos para los siguientes lenguajes sobre el alfabeto $\{a, b, c\}$:

1. L_1 : palabras del lenguaje $(a + b)^*(b + c)^*$.

El AFND asociado a L_1 es el de la Figura 1.51.

El AFD minimal asociado a L_1 es el de la Figura 1.52.

Figura 1.51: AFND asociado a L_1 del Ejercicio 1.1.26.Figura 1.52: AFD minimal asociado a L_1 del Ejercicio 1.1.26.

2. L_2 : palabras en las que nunca hay una “a” posterior a una “c”.

Vemos que L_2 tiene el mismo AFD que el de la Figura 1.52, por lo que $L_1 = L_2$.

3. $(L_1 \setminus L_2) \cup (L_2 \setminus L_1)$

Como $L_1 = L_2$, tenemos que $(L_1 \setminus L_2) \cup (L_2 \setminus L_1) = \emptyset$.

Ejercicio 1.1.27. Si $f : \{0, 1\}^* \rightarrow \{a, b, c\}^*$ es un homomorfismo dado por

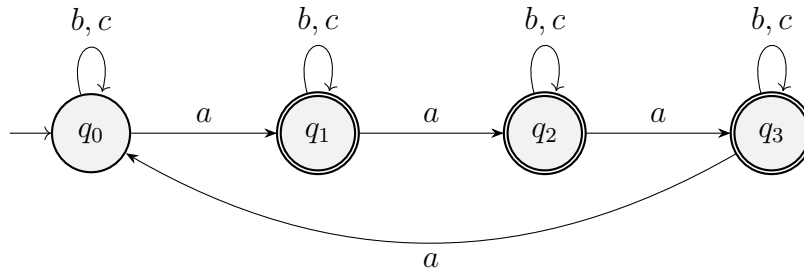
$$f(0) = aab \quad f(1) = bbc$$

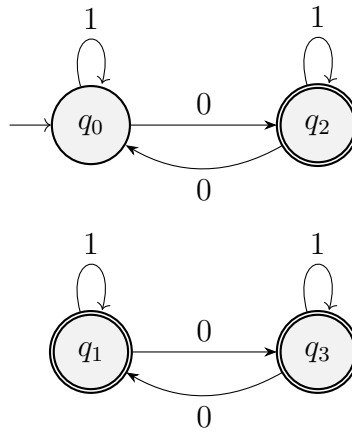
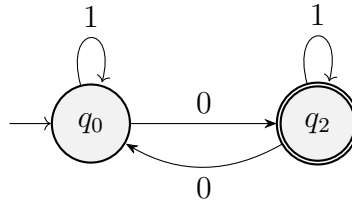
dar autómatas finitos deterministas minimales para los lenguajes L y $f^{-1}(L)$ donde $L \subseteq \{a, b, c\}^*$ es el lenguaje en el que el número de símbolos a no es múltiplo de 4.

El autómata finito minimal asociado a L es el de la Figura 1.53, donde q_i representa el estado en el que $n_a \bmod 4 = i$. Empleando un distinto número de cadenas de a 's, vemos que los estados son distinguibles, por lo que es minimal.

Empleando el algoritmo, el AFD asociado a $f^{-1}(L)$ es el de la Figura 1.54. No obstante, y debido a que hay estados inaccesibles, este no es minimal. El AFD minimal asociado a $f^{-1}(L)$ es el de la Figura 1.55.

Ejercicio 1.1.28. Si L_1 es el lenguaje asociado a la expresión regular $01(01 + 1)^*$ y L_2 el lenguaje asociado a la expresión $(1 + 10)^*01$, encontrar un autómata minimal

Figura 1.53: AFD minimal asociado a L del Ejercicio 1.1.27.

Figura 1.54: AFD asociado a $f^{-1}(L)$ del Ejercicio 1.1.27.Figura 1.55: AFD minimal asociado a $f^{-1}(L)$ del Ejercicio 1.1.27.

que acepte el lenguaje $L_1 \setminus L_2$.

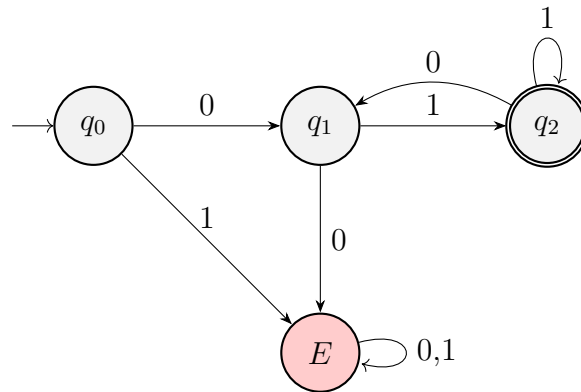
El AFD minimal asociado a L_1 es el de la Figura 1.56.

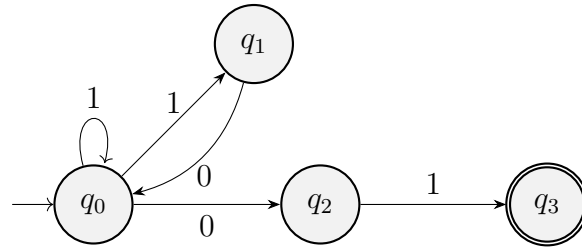
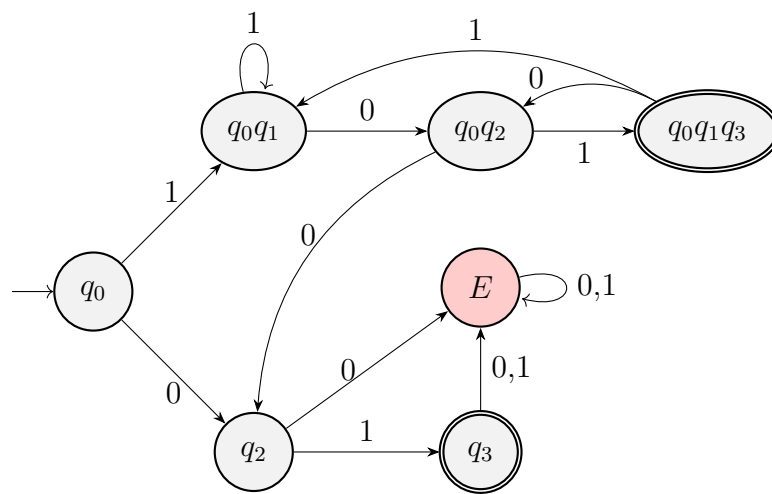
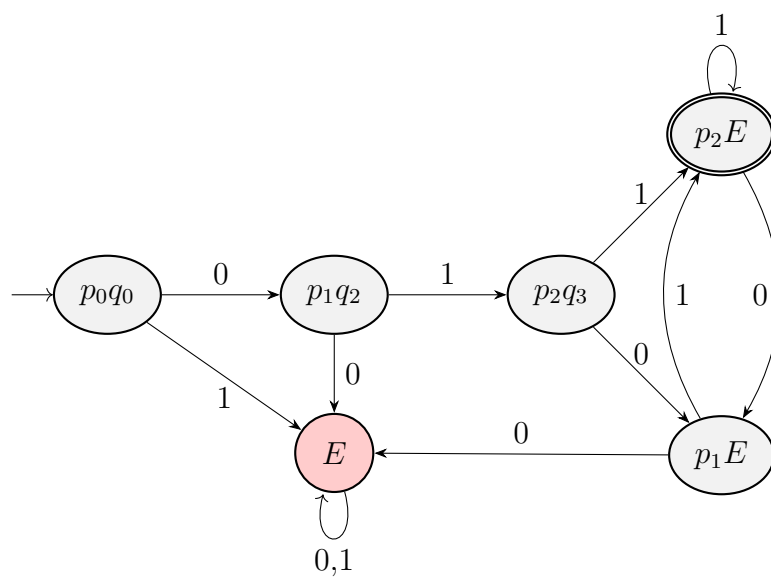
El AFND asociado a L_2 es el de la Figura 1.57.

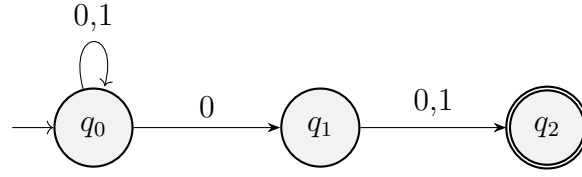
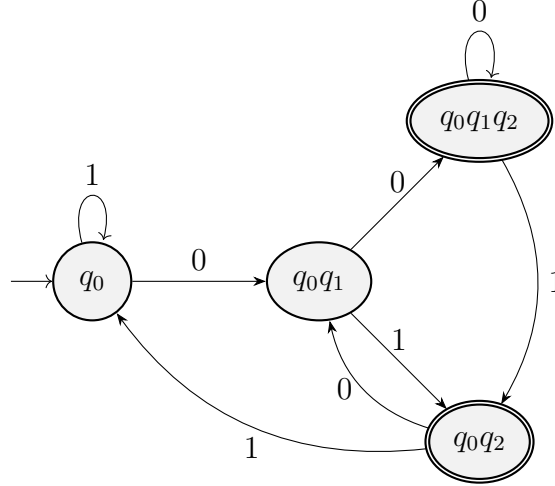
El AFD asociado a L_2 es el de la Figura 1.58.

El autómata asociado a $L_1 \setminus L_2$ es el de la Figura 1.59, donde hemos agrupado los estados de la forma E, q_i en E (ya que todos esos son indistinguibles puesto que no son finales y no se puede llegar a ellos desde un estado final).

Ejercicio 1.1.29. Sean los alfabetos $A_1 = \{a, b, c, d\}$ y $A_2 = \{0, 1\}$ y el lenguaje $L \subseteq A_2^*$ dado por la expresión regular $(0 + 1)^* 0(0 + 1)$, calcular una expresión regular

Figura 1.56: AFD minimal asociado a L_1 del Ejercicio 1.1.28.

Figura 1.57: AFND asociado a L_2 del Ejercicio 1.1.28.Figura 1.58: AFD minimal asociado a L_2 del Ejercicio 1.1.28.Figura 1.59: AFD minimal asociado a $L_1 \setminus L_2$ del Ejercicio 1.1.28.

Figura 1.60: AFND asociado a L del Ejercicio 1.1.29.Figura 1.61: AFD minimal asociado a L del Ejercicio 1.1.29.

para el lenguaje $f^{-1}(L)$ donde f es el homomorfismo entre A_1^* y A_2^* dado por

$$f(a) = 01 \quad f(b) = 1 \quad f(c) = 0 \quad f(d) = 00$$

El AFND asociado a L es el de la Figura 1.60.

El AFD minimal asociado a L es el de la Figura 1.61.

El AFD asociado a $f^{-1}(L)$ es el de la Figura 1.62.

Para obtener la expresión regular asociada a $f^{-1}(L)$, planteamos el sistema de ecuaciones siguiente:

$$\begin{cases} q_0 = bq_0 + cq_0q_1 + aq_0q_2 + dq_0q_1q_2 \\ q_0q_1 = (c+d)q_0q_1q_2 + (a+b)q_0q_2 \\ q_0q_1q_2 = (c+d)q_0q_1q_2 + (a+b)q_0q_2 + \varepsilon \\ q_0q_2 = cq_0q_1 + bq_0 + aq_0q_2 + dq_0q_1q_2 + \varepsilon \end{cases}$$

Sería necesario resolver el sistema para obtener la expresión regular.

Ejercicio 1.1.30. Obtener un autómata finito determinista para el lenguaje asociado a la expresión regular: $(01)^+ + (010)^*$. Minimizarlo.

El AFND asociado a la expresión regular es el de la Figura 1.63.

El AFD asociado a la expresión regular es el de la Figura 1.64.

La minimización del autómata se encuentra en la Tabla 1.15.

Por tanto, como todos los estados del AFD de la Figura 1.64 son distinguibles, este es minimal.

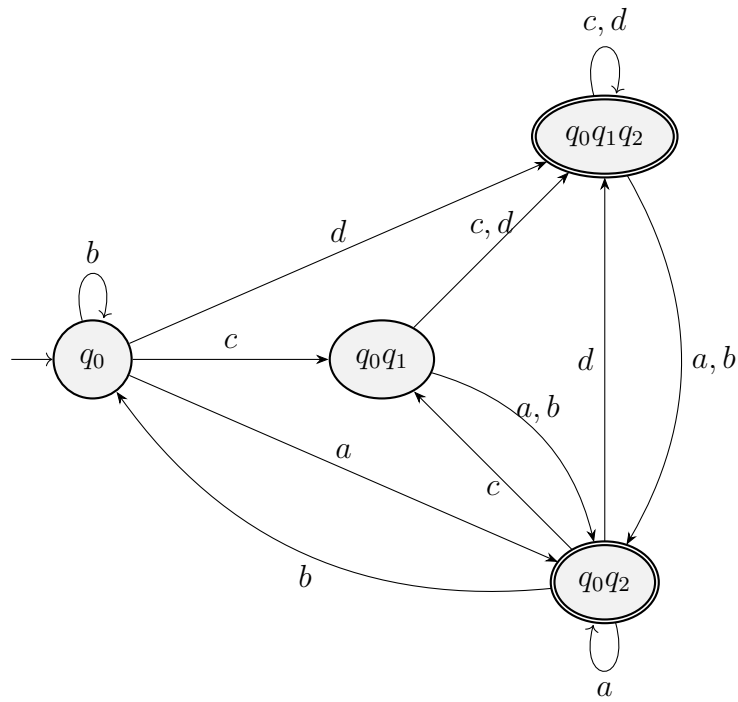
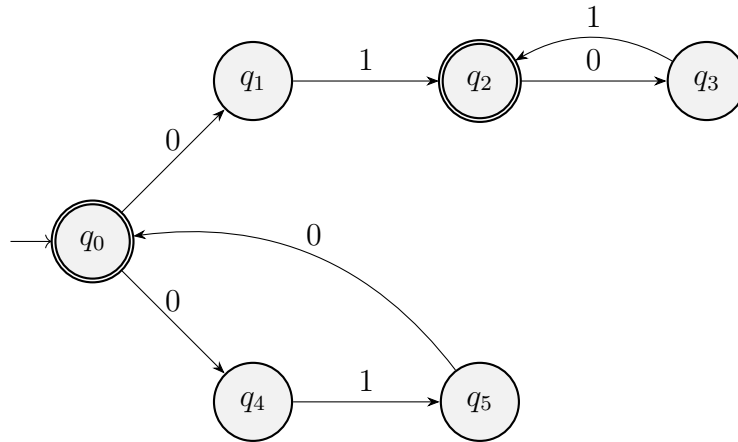
Figura 1.62: AFD asociado a $f^{-1}(L)$ del Ejercicio 1.1.29.

Figura 1.63: AFND asociado a la expresión regular del Ejercicio 1.1.30.

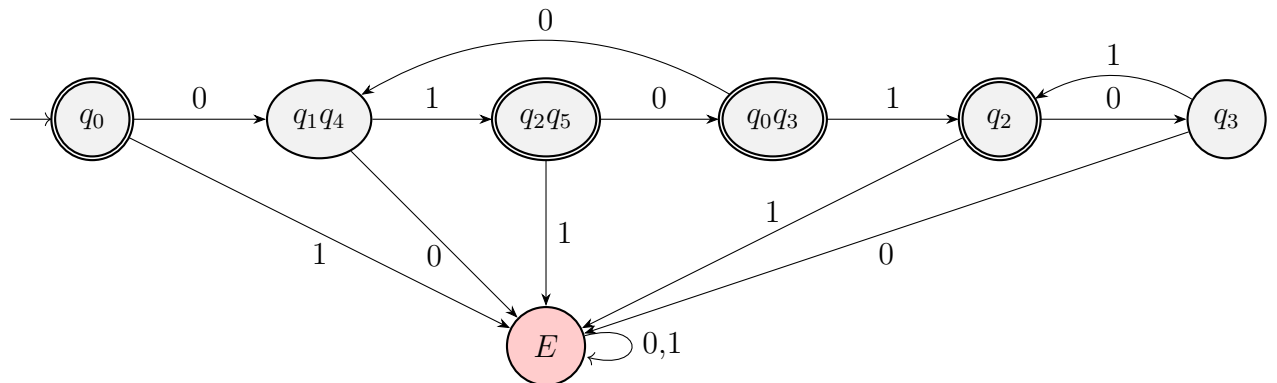
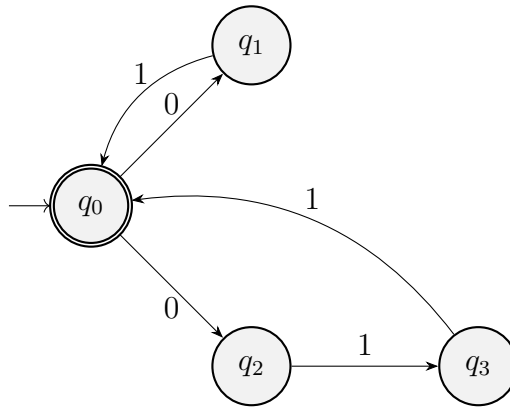


Figura 1.64: AFD asociado a la expresión regular del Ejercicio 1.1.30.

q_1q_4	×					
q_2q_5	×	×				
q_0q_3	×	×	×			
q_2	×	×	(q_3, q_1q_4)	×		
q_3	×	(q_0, q_2)	×	×	×	
E	×	×	×	×	×	×
	q_0	q_1q_4	q_2q_5	q_0q_3	q_2	q_3

Tabla 1.15: Minimización del autómata del Ejercicio 1.1.30.

Figura 1.65: AFND asociado al lenguaje L del Ejercicio 1.1.31.

Ejercicio 1.1.31. Dado el lenguaje L asociado a la expresión regular $(01 + 011)^*$ y el homomorfismo $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ dado por $f(0) = 01$, $f(1) = 1$, construir una expresión regular para el lenguaje $f^{-1}(L)$.

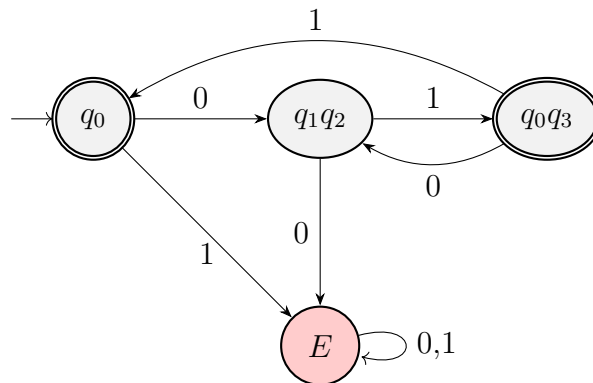
El AFND asociado a la expresión regular es el de la Figura 1.65.

El AFD asociado a la expresión regular es el de la Figura 1.66.

El AFD asociado a $f^{-1}(L)$ es el de la Figura 1.67.

No obstante, este no es minimal, puesto que tiene estados inaccesibles. El AFD minimal asociado a $f^{-1}(L)$ es el de la Figura 1.68.

Para obtener la expresión regular asociada a $f^{-1}(L)$, planteamos el sistema de

Figura 1.66: AFD asociado al lenguaje L del Ejercicio 1.1.31.

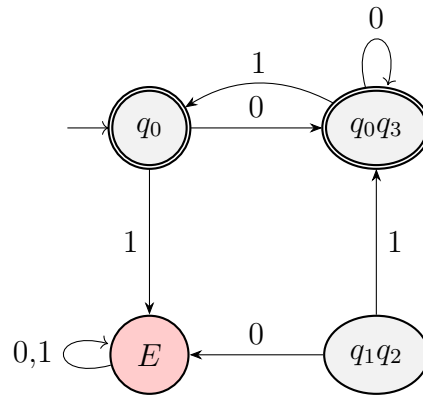


Figura 1.67: AFD asociado a $f^{-1}(L)$ del Ejercicio 1.1.31.

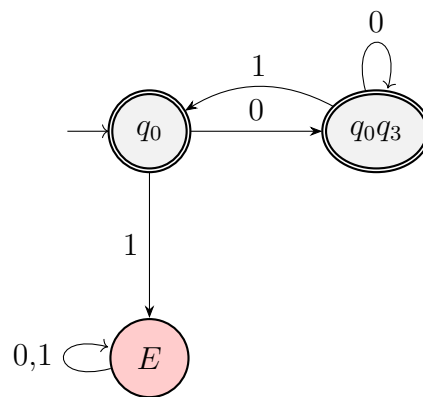
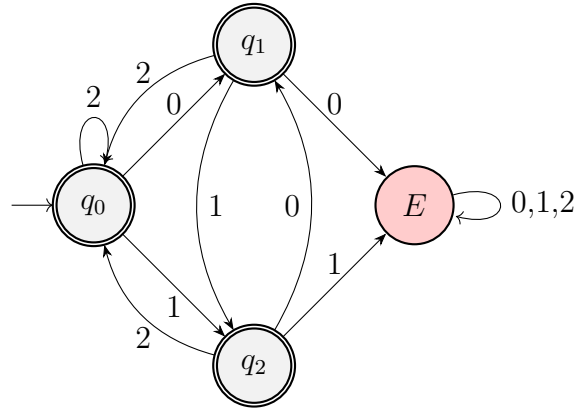


Figura 1.68: AFD minimal asociado a $f^{-1}(L)$ del Ejercicio 1.1.31.

Figura 1.69: AFD minimal asociado a L del Ejercicio 1.1.32.

ecuaciones siguiente:

$$\begin{cases} q_0 = 0q_0q_3 + 1E + \varepsilon \\ q_0q_3 = 0q_0q_3 + 1q_0 + \varepsilon \\ E = (0 + 1)E \end{cases}$$

Por el Lema de Arden, tenemos que $E = (0 + 1)^*\emptyset = \emptyset$ y $q_0q_3 = 0^*(1q_0 + \varepsilon)$. Por tanto, tenemos que:

$$\begin{aligned} q_0 &= 00^*(1q_0 + \varepsilon) + \varepsilon = 0^+(1q_0 + \varepsilon) + \varepsilon \implies \\ \implies q_0 &= (0^+1)^*(0^+ + \varepsilon) \end{aligned}$$

Por tanto, la expresión regular asociada a $f^{-1}(L)$ es $(0^+1)^*(0^+ + \varepsilon)$.

Ejercicio 1.1.32. Dar expresiones regulares para los siguientes lenguajes sobre el alfabeto $A_1 = \{0, 1, 2\}$:

1. L dado por el conjunto de palabras en las que cada 0 que no sea el último de la palabra va seguido por un 1 y cada 1 que no sea el último símbolo de la palabra va seguido por un 0.

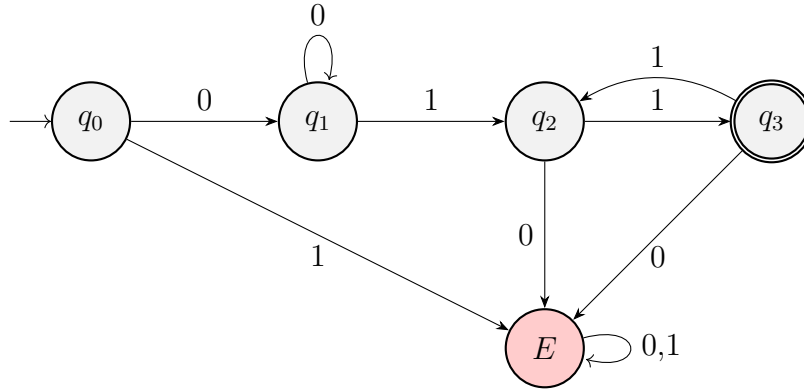
El AFD minimal asociado a L es el de la Figura 1.69.

Para obtener la expresión regular asociada a L , planteamos el sistema de ecuaciones siguiente:

$$\begin{cases} q_0 = 0q_1 + 1q_2 + 2q_0 + \varepsilon \\ q_1 = 1q_2 + 0E + 2q_0 + \varepsilon \\ q_2 = 1E + 0q_1 + 2q_0 + \varepsilon \\ E = (0 + 1 + 2)E \end{cases}$$

Por el Lema de Arden, tenemos que $E = (0 + 1 + 2)^*\emptyset = \emptyset$, por lo que $q_2 = 0q_1 + 2q_0 + \varepsilon$. Por tanto:

$$\begin{aligned} q_1 &= 1(0q_1 + 2q_0 + \varepsilon) + 0\emptyset + 2q_0 + \varepsilon = 1(0q_1 + 2q_0 + \varepsilon) + 2q_0 + \varepsilon \implies \\ \implies q_1 &= (10)^*(1 + (1 + \varepsilon)2q_0 + \varepsilon) = \\ &= (10)^*(1 + \varepsilon)(2q_0 + \varepsilon) \end{aligned}$$

Figura 1.70: AFD minimal asociado a L_1 del Ejercicio 1.1.33.

Por tanto, la expresión regular asociada a L es:

$$\begin{aligned}
 q_0 &= 0(10)^*(1 + \varepsilon)(2q_0 + \varepsilon) + 1(0(10)^*(1 + \varepsilon)(2q_0 + \varepsilon) + 2q_0 + \varepsilon) + 2q_0 + \varepsilon = \\
 &= 0(10)^*(1 + \varepsilon)(2q_0 + \varepsilon) + 10(10)^*(1 + \varepsilon)(2q_0 + \varepsilon) + 1(2q_0 + \varepsilon) + 2q_0 + \varepsilon = \\
 &= [0(10)^*(1 + \varepsilon) + 10(10)^*(1 + \varepsilon) + 1 + \varepsilon](2q_0 + \varepsilon) = \\
 &= [0(10)^* + 10(10)^* + \varepsilon](1 + \varepsilon)(2q_0 + \varepsilon) = \\
 &= [(1 + \varepsilon)0(10)^* + \varepsilon](1 + \varepsilon)(2q_0 + \varepsilon) = \\
 &= [[(1 + \varepsilon)0(10)^* + \varepsilon](1 + \varepsilon)2]^*[(1 + \varepsilon)0(10)^* + \varepsilon](1 + \varepsilon)
 \end{aligned}$$

2. Considera el homomorfismo de A_1 en $A_2 = \{0, 1\}$ dado por $f(0) = 001$, $f(1) = 100$, $f(2) = 0011$. Dar una expresión regular para $f(L)$.

La expresión regular asociada a $f(L)$ es:

$$[[(100 + \varepsilon)001(100001)^* + \varepsilon] (100 + \varepsilon)0011]^* [(100 + \varepsilon)001(100001)^* + \varepsilon] (100 + \varepsilon)$$

3. Dar una expresión regular para LL^{-1} .

Razonando los AFD vemos que $L = L^{-1}$, por lo que la expresión regular asociada a LL^{-1} es:

$$[[[(1 + \varepsilon)0(10)^* + \varepsilon](1 + \varepsilon)2]^* [(1 + \varepsilon)0(10)^* + \varepsilon](1 + \varepsilon)]^2$$

Ejercicio 1.1.33. Dados los lenguajes

$$L_1 = \{0^i 1^j \mid i \geq 1, j \text{ es par y } j \geq 2\}$$

$$L_2 = \{1^j 0^k \mid k \geq 1, j \text{ es impar y } j \geq 1\}$$

Encuentre:

1. Una gramática regular que genere el lenguaje L_1 .

El AFD minimal asociado a L_1 es el de la Figura 1.70.

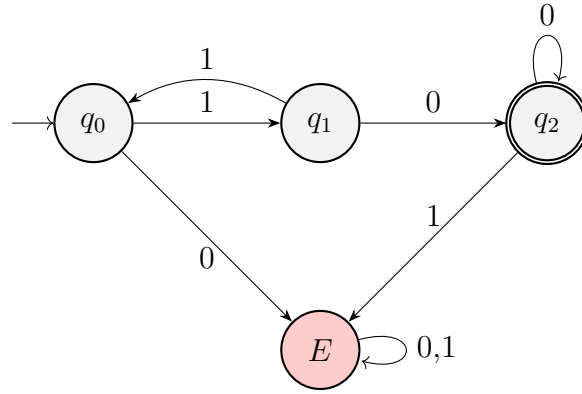


Figura 1.71: AFD minimal asociado a L_2 del Ejercicio 1.1.33.

La gramática regular asociada a L_1 por tanto que lo genera es $G = (V, \{0, 1\}, P, q_0)$ donde P es:

$$V = \{q_0, q_1, q_2, q_3\}$$

$$P = \begin{cases} q_0 \rightarrow 0q_1 \\ q_1 \rightarrow 0q_1 \mid 1q_2 \\ q_2 \rightarrow 1q_3 \\ q_3 \rightarrow 1q_2 \mid \varepsilon \end{cases}$$

2. Una expresión regular que represente al lenguaje L_2 .

El AFD minimal asociado a L_2 es el de la Figura 1.71.

La expresión regular asociada a L_2 se obtiene resolviendo el sistema de ecuaciones siguiente:

$$\begin{cases} q_0 = 1q_1 + 0E \\ q_1 = 1q_0 + 0q_2 \\ q_2 = 0q_2 + 1E + \varepsilon \\ E = (0 + 1)E \end{cases}$$

Por el Lema de Arden, tenemos que $E = (0 + 1)^*\emptyset = \emptyset$ y $q_2 = 0^*\varepsilon = 0^*$. Por tanto:

$$q_1 = 1q_0 + 00^+ = 1q_0 + 0^+$$

Por tanto, la expresión regular asociada a L_2 es:

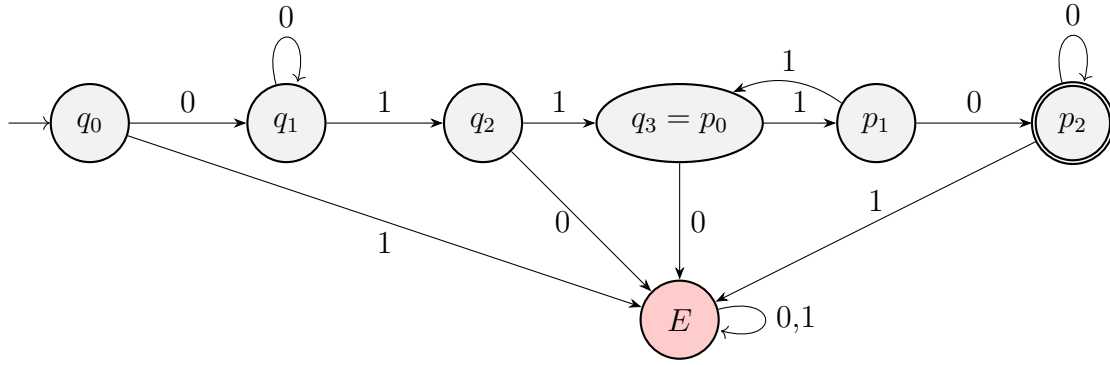
$$q_0 = 1(1q_0 + 0^+) = (11)^*(10^+)$$

3. Un automata finito determinista que acepte las cadenas de la concatenación de los lenguajes L_1 y L_2 . Aplica el algoritmo para minimizar este autómata.

El AFD asociado a L_1L_2 es el de la Figura 1.72. Este es:

$$\begin{aligned} L_1L_2 &= \{0^i1^{j+j'}0^k \mid i \geq 1, j \text{ es par y } j \geq 2, j' \text{ es impar y } j' \geq 1, k \geq 1\} = \\ &= \{0^i1^j0^k \mid i \geq 1, j \text{ es impar y } j \geq 3, k \geq 1\} \end{aligned}$$

Este autómata vemos de forma directa que es minimal, puesto que desde cada estado para llegar al único estado final hemos de leer una cadena distinta.

Figura 1.72: AFD asociado a L_1L_2 del Ejercicio 1.1.33.

δ_1	A	B	C	D	E	F	G	H
0	B	G	A	C	H	C	G	G
1	F	C	C	G	F	G	E	C

Tabla 1.16: Transiciones del autómata M_1 del Ejercicio 1.1.34.

Ejercicio 1.1.34. Considerar los AFD $M_1 = (\{A, B, C, D, E, F, G, H\}, \{0, 1\}, \delta_1, A, \{C\})$ y $M_2 = (\{A', B', C', D', G'\}, \{0, 1\}, \delta_2, A', \{D'\})$ donde δ_1 y δ_2 están definidas por las Tablas 1.16 y 1.17 respectivamente. Determinar si ambos autómatas finitos generan el mismo lenguaje.

Como M_1 tiene más estados, comenzamos minimizando este autómata. Veamos cuáles son sus estados accesibles:

$$\{A, B, F, G, C, E, H\} = Q \setminus \{D\}$$

Por tanto, tenemos que el único estado no accesible es D . La tabla de minimalización se encuentra en la Tabla 1.18.

Por tanto, notando por \equiv a la relación de indistinguibilidad, tenemos que:

$$H \equiv B \quad A \equiv E$$

Por tanto, el autómata minimal de M_1 es:

$$M_1^{\min} = (\{(AE), (BH), C, F, G\}, \{0, 1\}, \delta_1^{\min}, (AE), \{C\})$$

donde δ_1^{\min} está definida por la Tabla 1.19.

δ_2	A'	B'	C'	D'	G'
0	G'	B'	D'	A'	B'
1	C'	A'	B'	D'	D'

Tabla 1.17: Transiciones del autómata M_2 del Ejercicio 1.1.34.

B	\times					
C	\times	\times				
E		\times	\times			
F	\times	\times	\times	\times		
G	\times	\times	\times	\times	\times	
H	\times	(E, A)	\times	\times	\times	\times
	A	B	C	E	F	G

Tabla 1.18: Minimalización del autómata M_1 del Ejercicio 1.1.34.

δ_1^{\min}	(AE)	(BH)	C	F	G
0	(BH)	G	(AE)	C	G
1	F	C	C	G	(AE)

Tabla 1.19: Transiciones del autómata M_1^{\min} del Ejercicio 1.1.34.

Como podemos ver, M_1^{\min} y M_2 son isomorfos, con isomorfismo f dado por:

$$\begin{aligned}
 f((AE)) &= A' && \text{(inicial)} \\
 f(C) &= D' && \text{(final)} \\
 f((BH)) &= G' \\
 f(F) &= C' \\
 f(G) &= B'
 \end{aligned}$$

Por tanto, tenemos que $\mathcal{L}(M_1) = \mathcal{L}(M_2)$.

Ejercicio 1.1.35. Comprobar si los autómatas de las Figuras 1.73a y 1.73b generan el mismo lenguaje.

En primer lugar, vemos que q_3 es distinguible del resto, pues que es el único estado desde el cual no se puede llegar a un estado final. La tabla de minimalización de M_1 se encuentra en la Tabla 1.20.

Por tanto, el autómata minimal de M_1 es el de la Figura 1.74.

Como vemos, el autómata minimal de M_1 es isomorfo al AFD asociado a M_2 (ya que cuenta con un estado de error con las transiciones restantes) con isomorfismo f

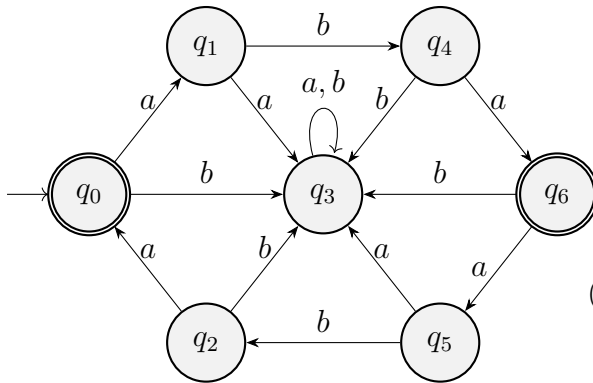
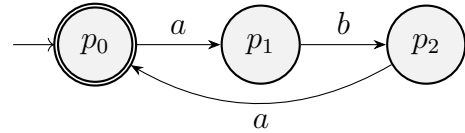
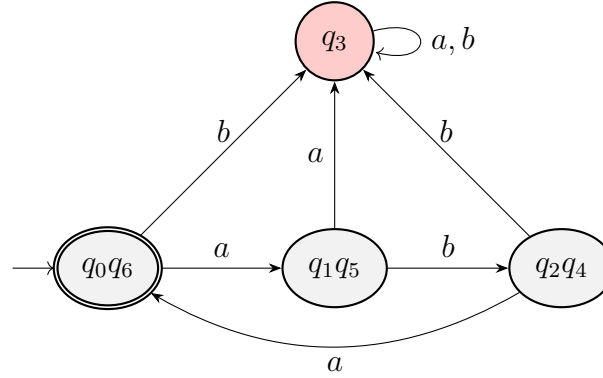
(a) Autómata M_1 del Ejercicio 1.1.35.(b) Autómata M_2 del Ejercicio 1.1.35.

Figura 1.73: Autómatas del Ejercicio 1.1.35.

q_1	×					
q_2	×	×				
q_3	×	×	×			
q_4	×	×	(q_1, q_5)	×		
q_5	×	(q_0, q_6)	×	×	×	
q_6	(q_2, q_4)	×	×	×	×	×
	q_0	q_1	q_2	q_3	q_4	q_5

Tabla 1.20: Minimalización del autómata M_1 del Ejercicio 1.1.35.Figura 1.74: Autómata minimal asociado a M_1 del Ejercicio 1.1.35.

dado por:

$$\begin{aligned}
 f(q_0q_6) &= p_0 && \text{(inicial)} \\
 f(q_1q_5) &= p_1 \\
 f(q_2q_4) &= p_2 && \text{(final)}
 \end{aligned}$$

Por tanto, tenemos que $\mathcal{L}(M_1) = \mathcal{L}(M_2)$.

Ejercicio 1.1.36. Minimizar el autómata de la Figura 1.75.

En primer lugar, vemos que q_7 es distinguible del resto, pues que es el único estado desde el cual no se puede llegar a un estado final. La tabla de minimalización de M se encuentra en la Tabla 1.21.

Por tanto, si notamos por \equiv a la relación de indistinguibilidad, tenemos que:

$$q_4 \equiv q_1 \quad q_5 \equiv q_2 \quad q_6 \equiv q_3$$

Por tanto, el autómata minimal de M es el de la Figura 1.76.

Ejercicio 1.1.37. Si L_1 y L_2 son lenguajes sobre el alfabeto A , entonces la *mezcla perfecta* de estos lenguajes se define como el lenguaje:

$$\{w \mid w = a_1b_1 \cdots a_kb_k \mid a_1 \dots a_k \in L_1, b_1 \dots b_k \in L_2, a_i, b_i \in A\}$$

Demostrar que si L_1 y L_2 son regulares, entonces la mezcla perfecta de L_1 y L_2 es regular.

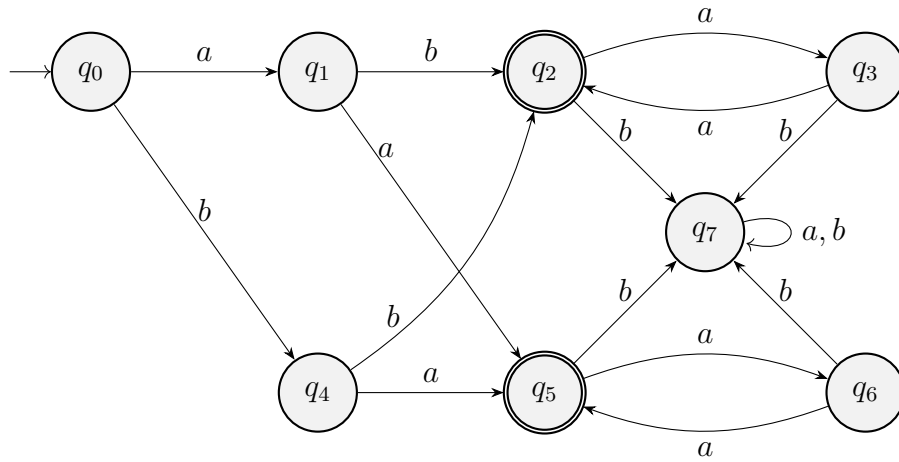
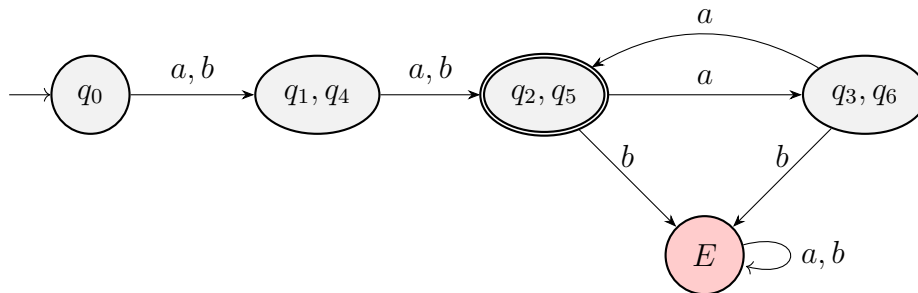


Figura 1.75: Autómata a minimizar del Ejercicio 1.1.36.

q_1	×						
q_2	×	×					
q_3	×	×	×				
q_4	×		×	×			
q_5	×	×	(q_3, q_6)	×	×		
q_6	×	×	×	(q_2, q_5)	×	×	
q_7	×	×	×	×	×	×	×
	q_0	q_1	q_2	q_3	q_4	q_5	q_6

Tabla 1.21: Minimalización del autómata M del Ejercicio 1.1.36.Figura 1.76: Autómata minimal asociado a M del Ejercicio 1.1.36.

Vamos a definir un autómata finito determinista que acepte la mezcla perfecta de L_1 y L_2 . Sean los autómatas $M_1 = (Q_1, A, \delta_1, q_0^1, F_1)$ el autómata asociado a L_1 y $M_2 = (Q_2, A, \delta_2, q_0^2, F_2)$ el autómata asociado a L_2 . Definimos el autómata $M = (Q, A, \delta, q_0, F)$ como sigue:

$$\begin{aligned} Q &= Q_1 \times Q_2 \times \{\mathcal{L}_1, \mathcal{L}_2\} \\ F &= F_1 \times F_2 \times \{\mathcal{L}_1\} \\ q_0 &= (q_0^1, q_0^2, \mathcal{L}_1) \\ \delta((q_1, q_2, \mathcal{L}_1), u) &= (\delta_1(q_1, u), q_2, \mathcal{L}_2) \quad \forall u \in A \\ \delta((q_1, q_2, \mathcal{L}_2), u) &= (q_1, \delta_2(q_2, u), \mathcal{L}_1) \quad \forall u \in A \end{aligned}$$

Notemos que el estado $(q_i, q_j, \mathcal{L}_k)$ indica que en el autómata M_1 se está en el estado q_i , en el autómata M_2 se está en el estado q_j y, ahora mismo, debemos leer una palabra de L_k . Por tanto, el autómata M acepta la mezcla perfecta de L_1 y L_2 , por lo que es regular.

Ejercicio 1.1.38. Si L es un lenguaje, sea $L_{1/2}$ el conjunto de palabras que son las mitades de palabras de L y $L_{-1/3}$ el conjunto de palabras que son las dos terceras partes de palabras de L . Es decir:

$$\begin{aligned} L_{1/2} &= \{x \mid \exists y \in A^*, |x| = |y|, xy \in L\} \\ L_{-1/3} &= \{xz \mid \exists y \in A^*, |x| = |y| = |z|, xyz \in L\} \end{aligned}$$

Demostrar que si L es regular, entonces $L_{1/2}$ también lo es, pero que $L_{-1/3}$ no es necesariamente regular.

1.1.1. Preguntas Tipo Test

Se pide discutir la veracidad o falsedad de las siguientes afirmaciones:

1. El lema de bombeo puede usarse para demostrar que un lenguaje determinado es regular.

Falso, el lema de bombeo nos dice que si un lenguaje es regular, entonces este cumple una determinada propiedad. Podemos usar su contrarrecíproco para ver que si una palabra del lenguaje no cumple dicha propiedad entonces el lenguaje no es regular, pero no nos sirve para determinar si un lenguaje lo es o no.

2. Todo lenguaje con un número finito de palabras es regular.

Verdadero, ya que si tenemos $L = \{v_1, v_2, \dots, v_n\}$ un lenguaje finito de $n \in \mathbb{N}$ palabras, entonces podemos construir la gramática $G = (\{S\}, A, P, S)$ con A el alfabeto sobre el que está definido L y P el siguiente conjunto de producciones:

$$P = \{S \rightarrow v_1 \mid v_2 \mid \dots \mid v_n\}$$

Con lo que G es una gramática regular de forma que $L = L(G)$, con lo que es regular.

3. La intersección de lenguajes regulares es siempre regular.

Verdadero. Como hemos visto en teoría, si tenemos dos lenguajes regulares, entonces podemos construir un autómata finito determinista para cada uno de ellos y construir el autómata producto, que genera la intersección de ambos lenguajes, con lo que el lenguaje como resultado de intersecar los dos lenguajes es regular.

4. La demostración del lema de bombeo se basa en que si leemos una palabra de longitud mayor o igual al número de estados del autómata, entonces en el camino que se recorre en el diagrama de transición se produce un ciclo.

Verdadero, si tenemos un autómata finito determinista de n estados que reconoce un lenguaje regular, si leemos una palabra de longitud m con $m \geq n$, entonces en el “camino de lectura” de la palabra, hemos de pasar por $m + 1$ estados, con lo que pasaremos por al menos un estado dos veces o más, con lo que el autómata tendrá un ciclo.

5. Es más fácil determinar si una palabra pertenece a un lenguaje regular cuando éste viene dado por una expresión regular que cuando viene dado por un autómata finito determinista.

Falso, si tenemos un autómata finito determinista que acepta el lenguaje en cuestión, ver si la palabra está en el lenguaje será tan sencillo como ir realizando las transiciones entre estados en el autómata leyendo la palabra y comprobando si al final llegamos a un estado final o si no. Por otra parte, puede suceder que ver si una palabra está en un lenguaje o no a partir de la expresión regular puede que sea sencillo, pero si consideramos una expresión regular muy compleja, posiblemente no sea tan fácil determinar si la palabra está en el lenguaje o no. En cualquier caso, reconocer si una palabra está en el lenguaje por un autómata finito determinista es siempre el proceso más sencillo.

6. En la demostración de que todo autómata finito tiene una expresión regular que representa el mismo lenguaje, el conjunto R_{ij}^k se define como el lenguaje de todas las palabras que llevan al autómata del estado q_i al estado q_j pasando por el estado número k , q_k .

Falso, (se trata de una pregunta del Tema 2) ya que R_{ij}^k se define como el conjunto de todas las palabras que llevan al autómata del estado q_i al estado q_j pasando por estados de numeración menor o igual que k , pero no necesariamente por q_k .

7. El conjunto de todas las expresiones regulares es un lenguaje regular.

Falso, si consideramos que la buena parentización¹ forma parte de las expresiones regulares, podemos demostrar que el conjunto de todas las expresiones regulares no es un lenguaje regular usando el Lema de bombeo:

Sea $n \in \mathbb{N}$ y $a \in A$, consideramos² $z = ({}^n a[+a])^n$. Es decir:

$$z = (((\dots(((a + a) + a) + a) \dots + a) + a) + a)$$

donde hay n paréntesis de apertura y n paréntesis de cierre y notemos que $|z| \geq n$. Supongamos que hay $u, v, w \in A^*$ tales que $z = uvw$ con $|v| \geq 1$ y $|uv| \leq n$. Entonces, $u = ({}^k$, $v = ({}^h$ con $h \geq 1$, $k + h \leq n$ y:

$$w = ({}^{n-k-h} a + a) + a) + a) \dots + a) + a) + a) = ({}^{n-k-h} a[+a])^n$$

Sin embargo, $uv^0w = uw = ({}^{n-h} a[+a])^n$, que no es una expresión regular correcta por no estar bien parentizada ($k \geq 1$), con lo que dicho lenguaje no es regular.

8. A partir de la demostración de que si R es regular y L un lenguaje cualquiera, entonces R/L es regular, se puede obtener un algoritmo para construir el autómata asociado a R/L .

Falso en general: si el lenguaje L es finito o regular, sí que nos dice cómo podemos construir el autómata asociado a R/L , pero si no puede suceder que no seamos capaces de hacerlo, por tener L infinitas palabras y no poder considerar en el autómata todas ellas.

9. En un autómata finito no-determinista, si intercambio entre sí los estados finales y no finales obtengo un autómata que acepta el lenguaje complementario.

Falso, ya que un autómata finito no determinista puede que no tenga definidas alguna transición desde un estado leyendo algún carácter (lo que en el determinista asociado significaría ir a un estado de error), con lo que al realizar dicho cambio en el autómata no determinista, no consideramos dichas transiciones. Para ver esto más claro, consideramos el autómata finito no determinista de la Figura 1.77, que acepta el lenguaje:

$$L = \{01u \mid u \in \{0, 1\}^*\}$$

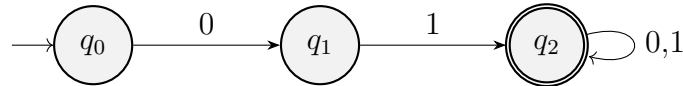


Figura 1.77: Autómata finito no determinista para la pregunta 9.

Si realizamos el cambio mencionado en la pregunta, entonces consideramos ahora como conjunto de estados finales $F' = \{q_0, q_1\}$, con lo que según esta pregunta, el nuevo autómata debería reconocer el lenguaje:

$$\bar{L} = \{w \mid w \neq 01u \quad \forall u \in \{0, 1\}^*\}$$

¹El poner paréntesis.

²Donde hemos usado corchetes para diferenciarlos de los paréntesis del lenguaje.

Sin embargo, $110 \in \bar{L}$ y si intentamos leer esta palabra en el nuevo autómata finito no determinista, no la reconoce, como resultado de la falta de transiciones en el autómata de la Figura 1.77, como habíamos enunciado anteriormente.

10. Si en un autómata finito no hay estados distinguibles de nivel 2, ya no puede haber estados distinguibles de nivel 4.
11. Todo lenguaje generado por una gramática lineal por la derecha es también generado por una gramática lineal por la izquierda.

Verdadero, se vió en el Tema 2.

12. Un autómata finito determinista sin estados inaccesibles ni indistinguibles es minimal.

Verdadero, gracias a un resultado visto en teoría.

13. Si L es un lenguaje sobre el alfabeto A , entonces $CAB(L)$ es siempre igual al cociente L/A^* .

Verdadero, por definición de $CAB(L)$ y de L/A^* :

$$L/A^* = \{u \in A^* \mid \exists v \in A^* \text{ verificando } uv \in L\} = CAB(L)$$

14. El lenguaje de las palabras sobre $\{0, 1\}$ en las que la diferencia entre el número de ceros y unos es impar es regular.

Verdadero, ya que se puede reconocer por el autómata de la Figura 1.78.

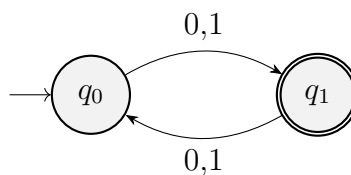


Figura 1.78: Autómata finito determinista para la pregunta 14.

Donde usamos el estado q_0 para representar que la diferencia entre el número de ceros y unos es par y q_1 para representar que la diferencia entre el número de ceros y unos es impar.

15. En un autómata finito cualquiera, si las transiciones dan lugar a un ciclo, entonces el lenguaje aceptado es infinito.

Falso, solo será cierto si tras salir de dicho ciclo se puede llegar a un estado final. Para ilustrar este caso, observamos el autómata finito determinista de la Figura 1.79.

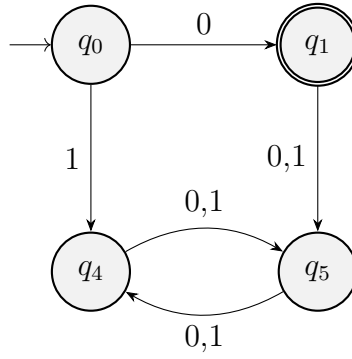


Figura 1.79: Autómata finito determinista para la pregunta 15.

En el autómata hay presente un ciclo y este reconoce el lenguaje $\{0\}$.

16. La expresión recursiva que se emplea para obtener la expresión regular asociada a un autómata finito determinista es: $r_{ij}^k = r_{ij}^{k-1} + r_{i(k-1)}^{k-1} (r_{(k-1)(k-1)}^{k-1})^* r_{(k-1)j}^{k-1}$.

Falso, aunque se trata de una pregunta del Tema 2, la expresión correcta es:

$$r_{ij}^k = r_{ij}^{k-1} + r_{ik}^{k-1} (r_{kk}^{k-1})^* r_{kj}^{k-1}$$

17. Cuando se construye la expresión regular asociada a un autómata finito determinista, r_{ii}^0 no puede ser nunca vacío.

Falso, sí puede serlo.

18. El conjunto de las palabras $\{u0011v^{-1} \mid u, v \in \{0, 1\}^*\}$ es regular.

Verdadero, para verlo más claro:

$$\{u0011v^{-1} \mid u, v \in \{0, 1\}^*\} = \{u0011w \mid u, w \in \{0, 1\}^*\}$$

Y podemos reconocer este lenguaje mediante la gramática lineal por la derecha $G = (\{S, A\}, \{0, 1\}, P, S)$ con P el conjunto que contiene las siguientes producciones:

$$\begin{aligned} S &\rightarrow 0S \mid 1S \mid 0011A \\ A &\rightarrow 0A \mid 1A \mid \varepsilon \end{aligned}$$

19. Si L es un lenguaje finito, entonces su complementario es siempre regular.

Verdadero, ya que si L es finito, entonces es regular, con lo que podemos construir un autómata finito determinista que reconozca dicho lenguaje. Una vez que obtengamos dicho autómata finito **determinista** $M = (Q, A, \delta, q_0, F)$, bastará considerar el autómata $M' = (Q, A, \delta, q_0, Q \setminus F)$, autómata finito determinista que aceptará el lenguaje $L(M') = \overline{L}$.

20. En un autómata finito determinista la relación de indistinguibilidad es una relación de equivalencia.

Verdadero, cumple las propiedades reflexiva, simétrica y transitiva, tal y como se ha visto en teoría.

21. En un autómata finito determinista siempre debe de existir, al menos, un estado de error.

Falso, el autómata que hicimos para la pregunta 14 que podemos ver en la Figura 1.78 era un autómata finito determinista totalmente válido y no tenía estados de error.

22. El conjunto de los números en binario que son múltiplos de 7 es regular.

Verdadero, ya que podemos construir un autómata finito determinista que reconozca dicho lenguaje, como vemos en la Figura 1.80, donde hemos hecho uso de la tabla de estados 1.22:

$7n$	$7n + 1$	$7n + 2$	$7n + 3$	$7n + 4$	$7n + 5$	$7n + 6$
0	1	2	3	4	5	6

Tabla 1.22: Estados en relación a si el número es múltiplo de 7 o si no.

Y de las propiedades de los números binarios:

- Leer un 0 es equivalente a multiplicar el número por 2.
- Leer un 1 es equivalente a multiplicar el número por 2 y sumarle 1.

De esta forma, conseguimos el autómata finito determinista anteriormente mencionado, razonando los pasos entre estados.

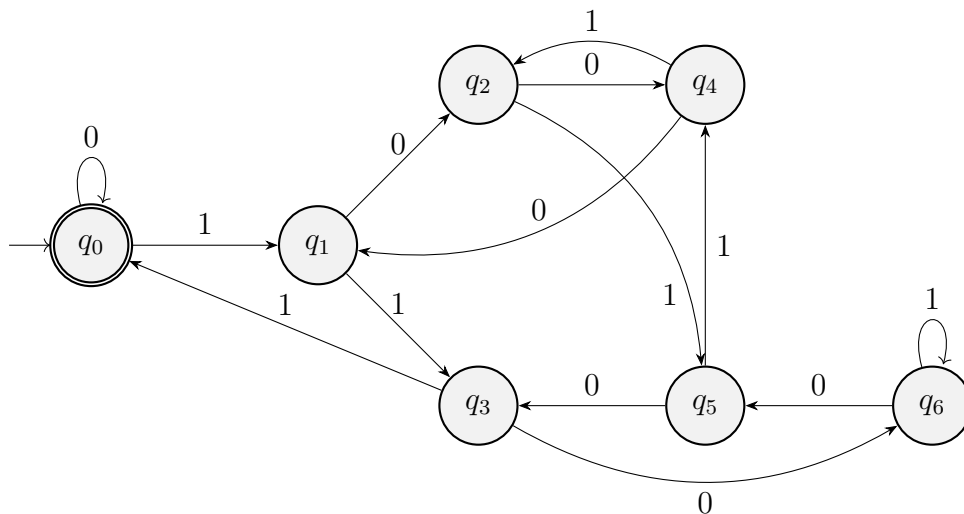


Figura 1.80: Autómata finito determinista para la pregunta 22.

23. Hay situaciones en las que los estados inaccesibles de un AFD cumplen una función específica.

Falso, no se puede llegar nunca a un estado inaccesible, por lo que son irrelevantes en el reconocimiento de una palabra.

24. Si R es un lenguaje regular y L un lenguaje independiente del contexto, entonces R/L es regular.

Verdadero, ya que no hace falta exigir hipótesis sobre L , sea cual sea dicho lenguaje, mientras que R sea regular, R/L será regular.

25. Si en un autómata dos estados son distinguibles de nivel n , entonces serán distinguibles de nivel m para todo $m \geq n$.

Verdadero, ya que dos estados son distinguibles de nivel n si y solo si existe una palabra $u \in A^*$ de longitud menor o igual que n tal que en el conjunto $\{\delta^*(p, u), \delta^*(q, u)\}$ hay un estado final y otro no final, por lo que si p y q son distinguibles de nivel n por la dicha existencia de una palabra $u \in A^*$, entonces serán indistinguibles de nivel m para $m \geq n$, ya que podemos considerar la misma palabra que considerábamos para el caso de n , por ser $|u| \leq n \leq m$.

26. Si h es un homomorfismo y $h(L)$ no es regular, podemos concluir que L no es regular.

Verdadero, es la implicación contrarrecíproca de “si h es un homomorfismo y L es regular, entonces $h(L)$ es regular”, vista en teoría.

27. El lenguaje de todas las palabras en las que los tres primeros símbolos son iguales a los tres últimos es regular.

Verdadero, supongamos que trabajamos sobre el alfabeto $A = \{a_1, \dots, a_n\}$. Sabemos que hay un número finito de combinaciones para coger tres símbolos de dicho lenguaje: n^3 combinaciones distintas. Podemos pues, construir una aplicación biyectiva $f : \{1, 2, \dots, n^3\} \rightarrow A \times A \times A$. De esta forma, podemos considerar la gramática:

$$G = (\{S\} \cup \{A_i\}_{i \in \{1, 2, \dots, n^3\}}, A, P, S)$$

Siendo P el conjunto de producciones que contienen las siguientes producciones que vamos a describir.

S va a poder generar cada una de las n^3 sucesiones de 3 símbolos sobre A , cada una seguida de una variable A_i siendo i el índice de dicha combinación. Posteriormente, todas las variables A_i podrán generar un número indefinido de ceros y unos en cualquier orden, teniendo que terminar con la combinación

de los 3 símbolos correspondiente al índice i :

$$\begin{aligned}
 S &\rightarrow f(1)A_1 \mid f(2)A_2 \mid \dots \mid f(n^3)A_{n^3} \\
 A_1 &\rightarrow 0A_1 \mid 1A_1 \mid f(1) \\
 A_2 &\rightarrow 0A_2 \mid 1A_2 \mid f(2) \\
 &\vdots \\
 A_i &\rightarrow 0A_i \mid 1A_i \mid f(i) \\
 &\vdots \\
 A_{n^3} &\rightarrow 0A_{n^3} \mid 1A_{n^3} \mid f(n^3)
 \end{aligned}$$

Y tenemos que G es una gramática regular por la derecha, por lo que el lenguaje que genera es regular.

28. Si un lenguaje verifica la condición que aparece en el lema de bombeo para lenguajes regulares, ya no hay forma de demostrar que no es regular.

Falso, el lenguaje puede verificar la condición del lema de bombeo y no ser regular.

29. Si f es un homomorfismo entre alfabetos $f : A_1^* \rightarrow A_2^*$ y $L \subseteq A_1^*$ no es regular, podemos concluir que $f(L)$ tampoco es regular.

30. Todo lenguaje que cumple la condición del lema de bombeo para lenguajes regulares puede ser aceptado por un autómata finito no determinista.

Falso, anteriormente comentamos que un lenguaje puede verificar la condición del lema de bombeo y no ser regular.

31. No existe algoritmo para saber si el lenguaje generado por una gramática regular es finito.

32. Dos autómatas finitos deterministas con diferente número de estados y que aceptan el lenguaje vacío tienen el mismo número de estados finales.

Falso, solo sería cierto si todos los estados son alcanzables, ya que los autómatas de las Figuras 1.81 y 1.82 ambos aceptan el lenguaje vacío y el número de estados finales es distinto.

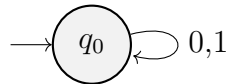


Figura 1.81: Autómata finito determinista 1 para la pregunta 32.

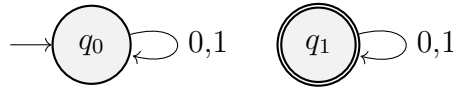


Figura 1.82: Autómata finito determinista 2 para la pregunta 32.

33. Si A es un alfabeto y L un lenguaje cualquiera distinto del vacío, entonces se verifica que $A^*/L = A^*$.

Verdadero, si recordamos la definición de A^*/L :

$$A^*/L = \{u \in A^* \mid \exists v \in L \text{ verificando } uv \in A^*\}$$

\subseteq Es trivial, por ser A^*/L un lenguaje.

\supseteq Sea $u \in A^*$, como $L \neq \emptyset$, existirá $v \in L$, con lo que $uv \in A^*$ por ser el conjunto de todas las palabras, con lo que $u \in A^*/L$.

34. Si R_{ij}^k son los lenguajes que se usan en la construcción de una expresión regular a partir de un autómata finito, siempre se verifica que $R_{ij}^{i-1} R_{jk}^{j-1} \subseteq R_{ik}^j$.
35. El lema de bombeo es útil para demostrar que la intersección de dos lenguajes regulares no es regular.
36. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es infinito.
37. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es finito o infinito.
38. La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.

Verdadero, ya que la intersección de dos lenguajes regulares da lugar a un lenguaje regular (tal y como veíamos en la pregunta 3), que a su vez es independiente del contexto.

39. Si un lenguaje es infinito no se puede encontrar una expresión regular que lo represente.

Falso, debido a la existencia de lenguajes regulares infinitos. Por ejemplo, podemos considerar $L = \{a^i \mid i \in \mathbb{N}\}$, un lenguaje infinito por ser biyectivo con \mathbb{N} :

$$L = \{\varepsilon, a, aa, aaa, aaaa, \dots\}$$

Y este puede ser representado por la expresión regular $(a)^*$.

40. En un autómata finito determinista sin estados inaccesibles la relación de indistinguibilidad entre los estados es una relación de equivalencia.

Verdadero, tal y como vimos en teoría.

41. En un autómata finito determinista, si no hay dos estados que sean indistinguibles entre sí, entonces el autómata es minimal.

Falso, salvo si el autómata no tiene estados inalcanzable, en cuyo caso es verdadero.

42. Dada una gramática lineal por la derecha, siempre existe otra gramática lineal por la izquierda que acepte el mismo lenguaje.

Verdadero, tal y como vimos en la teoría del Tema 2.

43. Si R es un lenguaje regular y L un lenguaje cualquiera, entonces R/L es siempre un lenguaje regular.

Verdadero, tal y como hemos visto en la teoría.

44. Si un lenguaje cumple la condición del lema de bombeo para conjuntos regulares no nos asegura que sea un lenguaje regular.

Verdadero, ya que la propiedad del lema de bombeo es una condición necesaria, no suficiente.

45. Existe un algoritmo para determinar si los lenguajes generados por dos gramáticas regulares son iguales o no.

Verdadero. En teoría hemos visto que si tenemos dos lenguajes, L_1 y L_2 generados por dos autómatas finitos de terministas, M_1 y M_2 (respectivamente), entonces podemos construir el autómata finito para el lenguaje $L_1 \setminus L_2$, y aplicarle el algoritmo de si el lenguaje que genera es vacío (en cuyo caso, $L_1 = L_2$). Además, como tenemos un algoritmo para pasar de gramáticas regulares a autómatas, podemos mecanizar todo este proceso.

46. El conjunto de cadenas aceptado por un autómata finito no determinista con transiciones nulas no puede ser generado por una gramática independiente del contexto.

Falso, el conjunto de cadenas aceptadas por un autómata finito no determinista con transiciones nulas es un lenguaje regular, que puede ser generado por una gramática regular por la derecha, que a su vez es independiente del contexto.

47. El lenguaje resultado de la unión de dos lenguajes regulares con un número infinito de palabras puede ser representado mediante una expresión regular.

Verdadero, ya que la unión de dos lenguajes regulares es regular, con lo que puede ser representado mediante una expresión regular, independientemente de la cardinalidad de los lenguajes.

48. Una expresión regular siempre representa a un lenguaje que puede ser generado por una gramática independiente del contexto.

Verdadero, ya que una expresión regular siempre representa a un lenguaje que puede ser generado por una gramática regular por la derecha, que a su vez es independiente del contexto.

49. Existe un algoritmo para comprobar si son iguales los lenguajes aceptados por dos autómatas finitos diferentes.

Verdadero, en caso de ser autómatas finitos deterministas, es el razonamiento que ya hicimos en la pregunta 45. En caso de ser autómatas finitos no deterministas, existe un algoritmo para pasarlos a deterministas y en cuyo caso, podemos aplicar el algoritmo ya mencionado.

50. Si en un autómata finito no determinista intercambio entre sí los estados finales y no finales obtengo un autómata que acepta el lenguaje complementario del aceptado por el autómata original.

Falso, esta pregunta ya fue respondida anteriormente, en la pregunta 9.

51. Si L es un lenguaje regular, entonces el lenguaje LL^{-1} es también regular.

Verdadero, ya que si L es regular, entonces L^{-1} es regular; y la concatenación de lenguajes regulares sigue siendo regular.

52. El lema de bombeo para lenguajes regulares es útil para demostrar que un lenguaje determinado no es regular.

Verdadero, puede ser útil para demostrar que un lenguaje no es regular, aunque puede haber casos en los que no nos dé información sobre si es regular o no, al tratarse de una condición necesaria para los lenguajes regulares.

53. Si un lenguaje tiene un conjunto infinito de palabras sabemos que no es regular.

Falso, el lenguaje $L = \{a^i \mid i \in \mathbb{N}\}$ tiene un conjunto infinito de palabras (es biyectivo con \mathbb{N}) y es regular, ya que puede representarse mediante la expresión regular $(a)^*$.

54. Un autómata finito determinista sin estados inaccesibles ni indistinguibles es minimal.

Verdadero, tal y como comentamos anteriormente en la pregunta 12.

55. El conjunto de las palabras $\{u0011v^{-1} \mid u, v \in \{0, 1\}^*\}$ es regular.

Verdadero, tal y como comentamos anteriormente en la pregunta 18.

56. Existe un algoritmo para determinar si el lenguaje generado por una gramática regular es infinito.
57. Para cada autómata finito no determinista M existe una gramática independiente de contexto G tal que $L(M) = L(G)$.

Verdadero, ya que sabemos que para cada autómata finito no determinista M existe una gramática regular por la derecha G tal que $L(M) = L(G)$ y sabemos que las gramáticas regulares por la derecha son a su vez independientes del contexto.

58. El lenguaje formado por las cadenas sobre $\{0, 1\}$ que tienen un número impar de 0 y un número par de 1 no es regular.

Falso, ya que podemos construir un autómata finito determinista que acepte el lenguaje, tal y como vemos en la Figura 1.83.

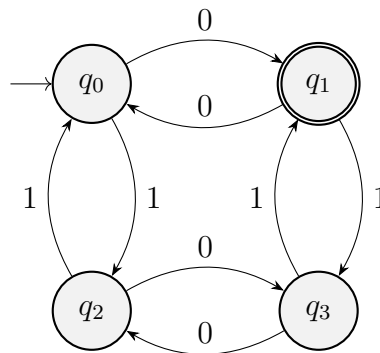


Figura 1.83: Autómata finito determinista para la pregunta 58.

Donde pensamos en los estados como:

- q_0 , la palabra tiene un número par de ceros y de unos.
- q_1 , la palabra tiene un número impar de ceros y número par de unos.
- q_2 , la palabra tiene un número par de ceros y número impar de unos.
- q_3 , la palabra tiene un número impar de ceros y de unos.

Elegimos como estado inicial q_0 ya que la palabra ε tiene un número par de ceros y de unos.

59. Si L es un lenguaje regular, entonces la cabecera de L ($CAB(L)$) es siempre regular.

Verdadero, si L es un lenguaje regular, existirá un autómata finito determinista $M = (Q, A, \delta, q_0, F)$ que reconozca dicho lenguaje. Consideramos ahora el autómata $M' = (Q, A, \delta, q_0, F')$, un autómata igual que M con la diferencia de que $q \in F'$ si y solo si existen dos palabras $u, v \in A^*$ tales que:

$$\delta^*(q_0, u) = q \quad \wedge \quad \delta^*(q, v) \in F$$

Es decir, si q es un estado alcanzable y se encuentra en el camino previo a un estado final, de forma que la palabra que se lee desde q_0 hasta q será un prefijo de alguna palabra de L .

60. En un autómata finito determinista, si no hay dos estados que sean indistinguibles entre si, entonces el autómata es minimal.

Falso, es necesario exigir además que no haya estados inalcanzables.

61. La intersección de dos lenguajes regulares da lugar a un lenguaje independiente del contexto.

Verdadero, ya que la intersección de dos lenguajes regulares da lugar a un lenguaje regular, que a su vez es independiente del contexto.

62. Si un lenguaje es infinito no se puede encontrar una expresión regular que lo represente.

Falso, pregunta que ya fue contestada en la pregunta 39.